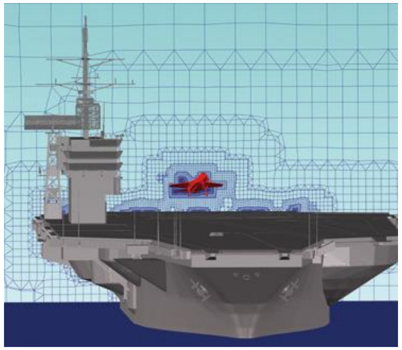
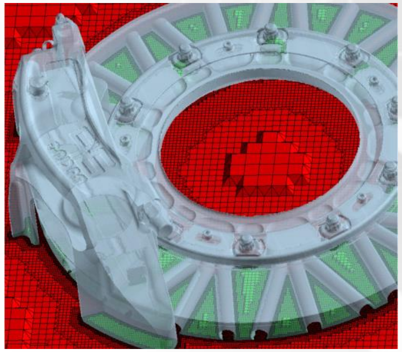
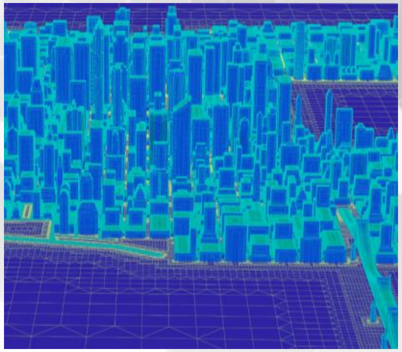
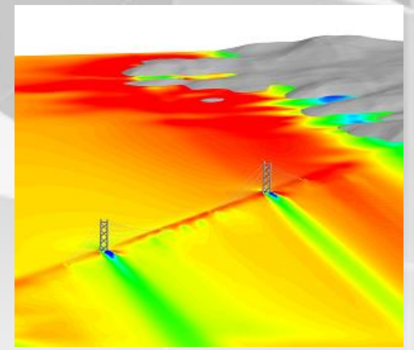
CAMBRIDGE FLOW SOLUTIONS

Proprietary and customised CFD

# PROGRESS TOWARDS AN INDUSTRY-FRIENDLY LES CAPABILITY

**2017 AIAA AVIATION Forum - CFD-19 Special Session**


Bill Dawes, Lu Yi, Kai Liu & Carlos Vasco

*Cambridge Flow Solutions Ltd., Cambridge UK*

*BOXER Solutions KK, Kobe, Japan*

# Overview

- Paper objective
- Why higher order ?
- The BOXER Environment & LES/*HOTnewt*
- Turbomachinery & aerospace examples
- Computer resources
- Post-processing
- Summary

# Paper objective

- Our objective is *not* to show LES and compare with experiment
  - lots of people do that
- Our objective is to show ***affordable*** LES which any engineer might perform at their desk
- This is delivered by a combination of novel software & hardware

CAMBRIDGE FLOW SOLUTIONS

# Overview

- Paper objective
- **Why higher order ?**
- The BOXER Environment & LES/*HOTnewt*
- Turbomachinery & aerospace examples
- Computer resources
- Post-processing
- Summary

# Why higher order ?

- There is widespread misunderstanding about the role & advantages of higher order methods
- The key measure of the usefulness of a simulation is the level of computer resource needed, in simple terms:
  - the power in kWh and elapsed wall-clock time needed to perform a simulation in which the appropriate physical space & time scales are successfully resolved
- Clearly higher order methods deliver higher accuracy than lower order methods on the same mesh but with more floating point operations needed to attain that accuracy
  - so the key question really is: which approach uses less resource:

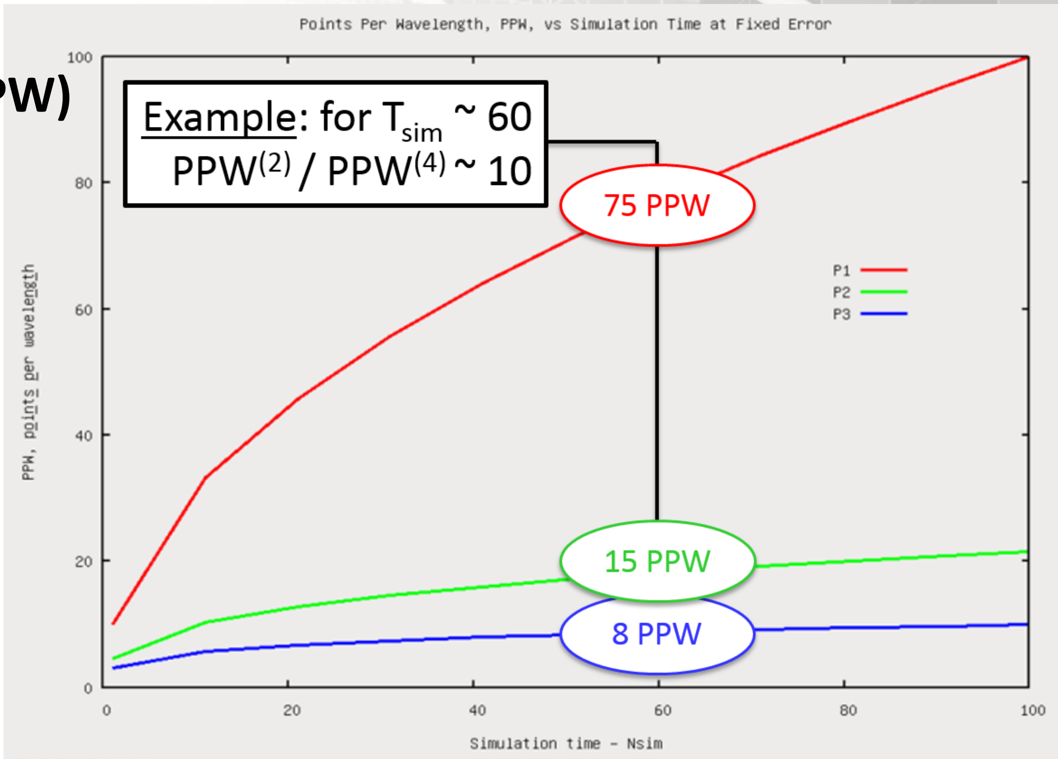
a low order method on a finer mesh or a high order method on a coarser mesh?

CAMBRIDGE FLOW SOLUTIONS

# Higher Order Simulation Methods – PPW [Leland Jameson, 2001]

**Points Per Wavelength (PPW)**

- The figure illustrates this trend for P1, P2 & P3…
- This shows that a low order P1 method ("second order") might need a factor of 10 more PPW in 1D than a P3 method ("fourth order") – *in 3D this becomes an astonishing factor of 1000 !*




**Simulation time $T_{sim}$ for fixed $E_{sim}$**

- Hence, ***provided the higher order method does not consume too much extra memory per DOF, or too many extra floating point operations, then a well-designed algorithm would permit a better computer memory – wall-clock performance***

CAMBRIDGE FLOW SOLUTIONS

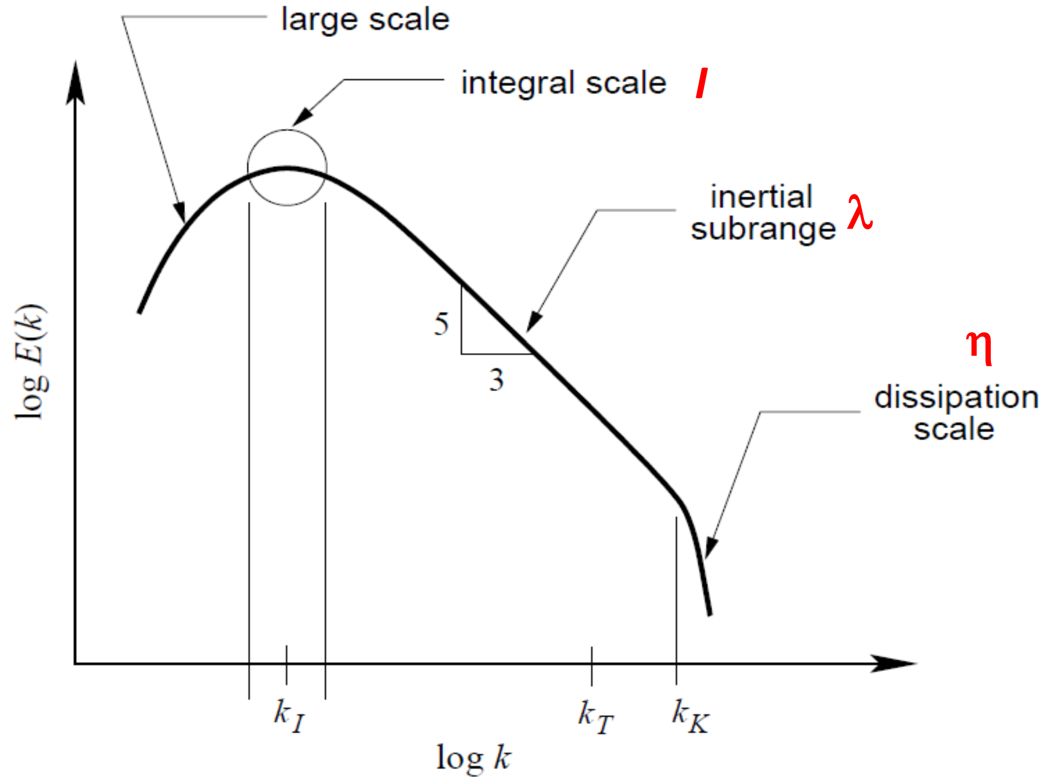
# Turbulence scales & the energy spectrum

- There are several scales
  - the **Integral scale** (or "outer" scale), *l*

    [commonly the scale is defined as *l* ~ 0.1 L where L is the scale of the flow domain]

  - the **Taylor microscale** within the inertial subrange, $\lambda$

  - the **Kolmogorov** (or "inner" scale), $\eta$

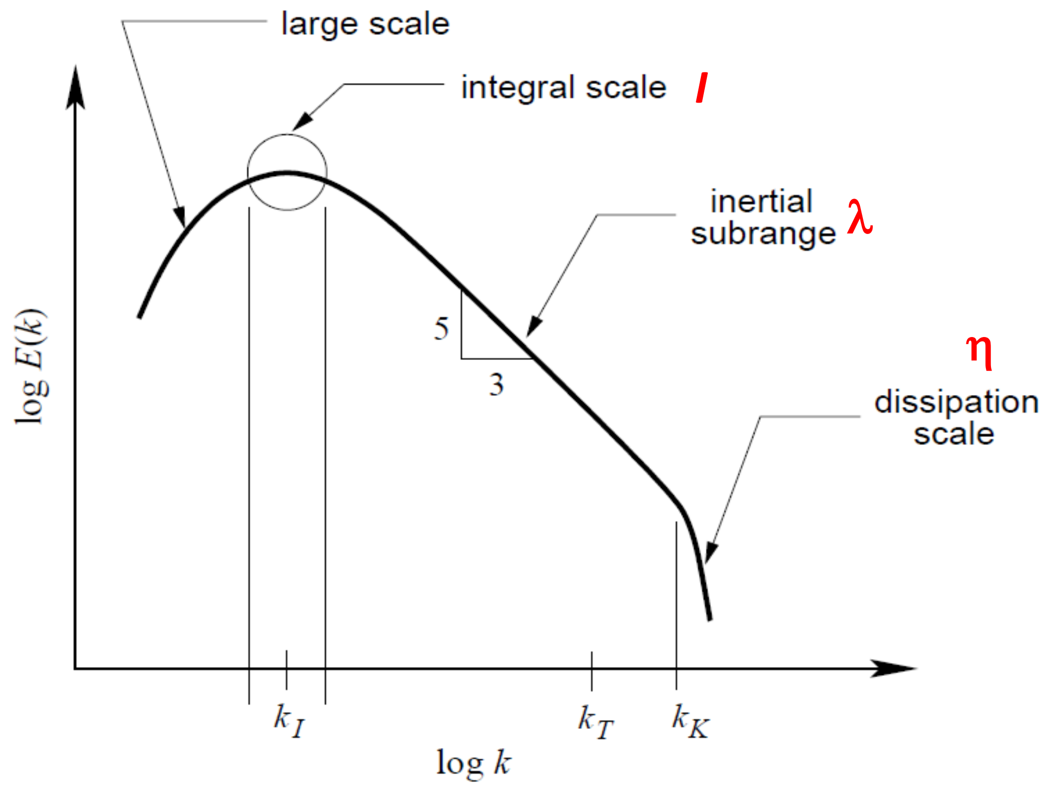



**Turbulence energy wavenumber spectrum**

# Turbulence scales & the energy spectrum

- These scales are related by:
  - $\eta/l \sim Re_l^{-3/4}$
  - $\lambda/l \sim Re_l^{-1/2}$

  where $Re_l = |u'|\,l/\nu$ is the "turbulent Reynolds number"
- Hence in terms of wavenumber:
  - Taylor scales: $k_T \sim k_l\, Re_l^{+1/2}$
  - Kolmogorov : $k_K \sim k_l\, Re_l^{+3/4}$




**Turbulence energy wavenumber spectrum**

CAMBRIDGE FLOW SOLUTIONS

# Example: LES resolution requirements

- Example:

  - Consider a LxLxL 3D domain with L=0.1m and $Re_L=10^{+6}$
  - then with $l=0.1L$

    and (say) $|u'|=10\%U$

    so that $Re_l=10^{+4}$

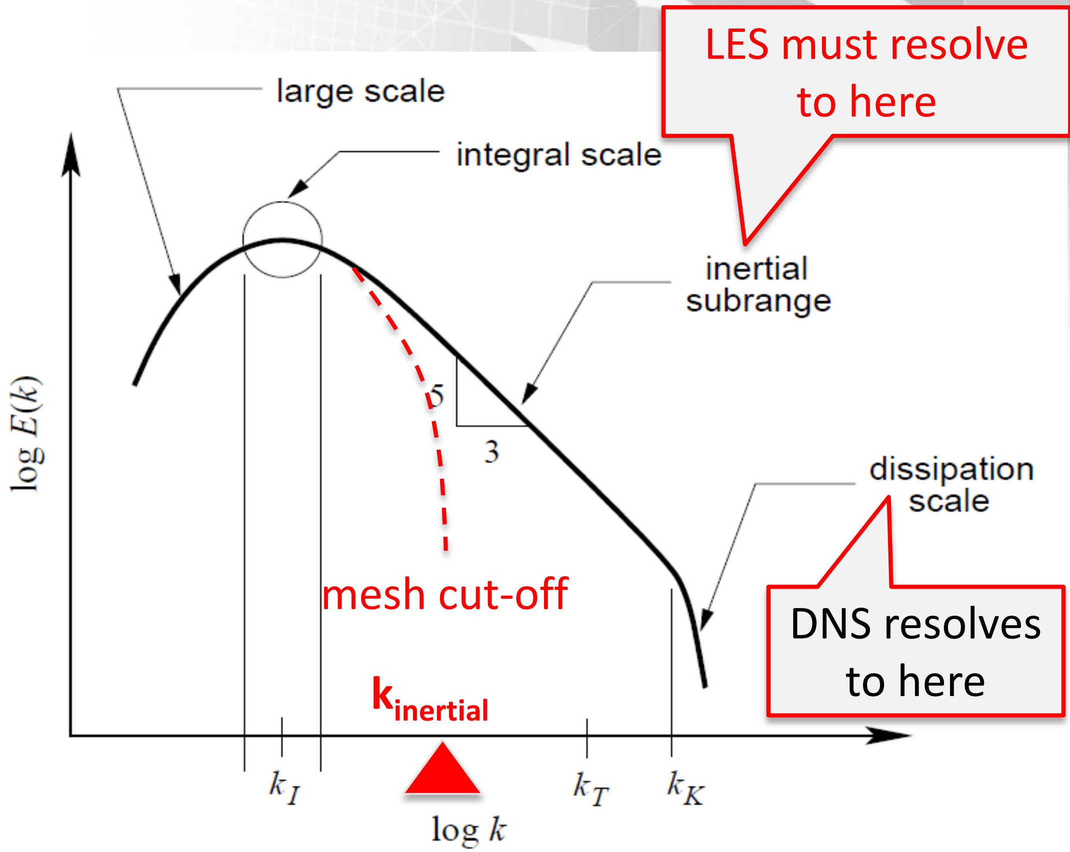
  - **Hence**

    Integral scale: $k_l \sim 314.$

    Taylor scale: $k_T \sim 31400.$

  - SO define a target wavenumber in the **inertial subrange**

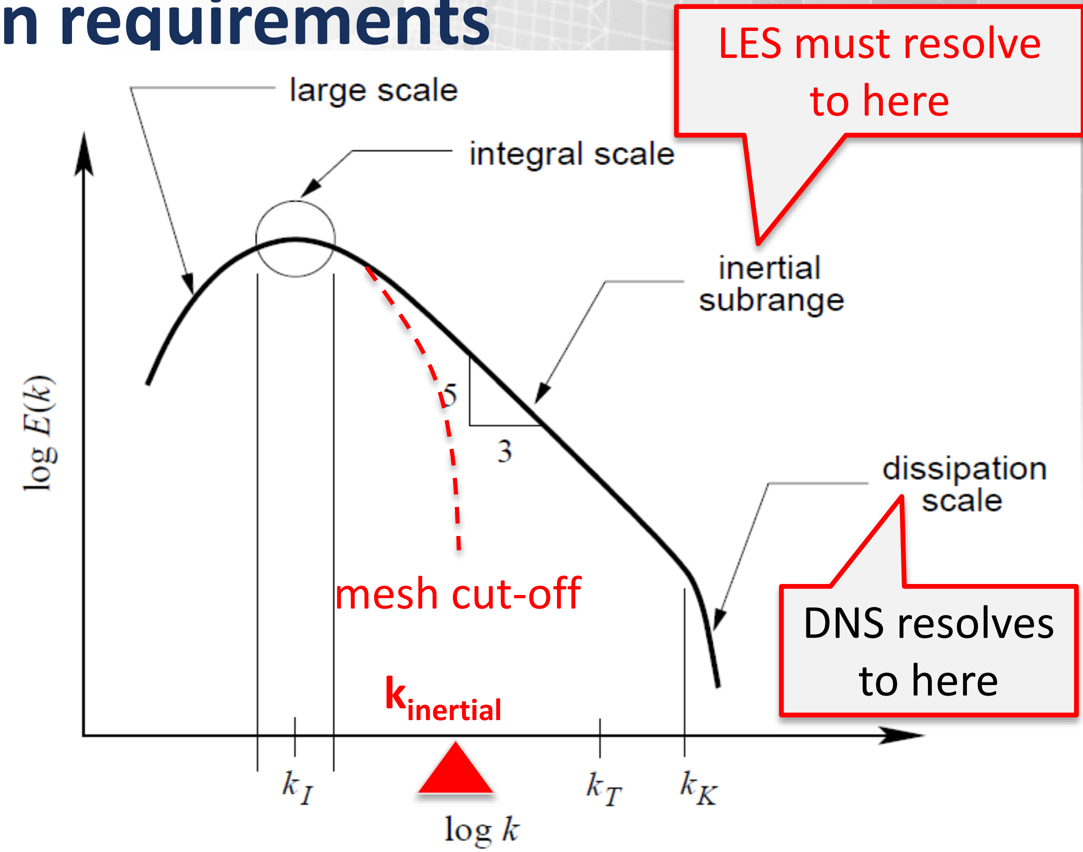
    $\mathbf{k_{inertial} \sim 3140.}$



# Example: LES resolution requirements

- The Table shows the mesh sizes that would be needed to resolve to the target wavenumber **$k_{inertial}$ ~ 3140.**

- Also shown are the total Degrees of Freedom (DOF) and an estimate of the floating point operations needed (from our own code – *HOTnewt*)

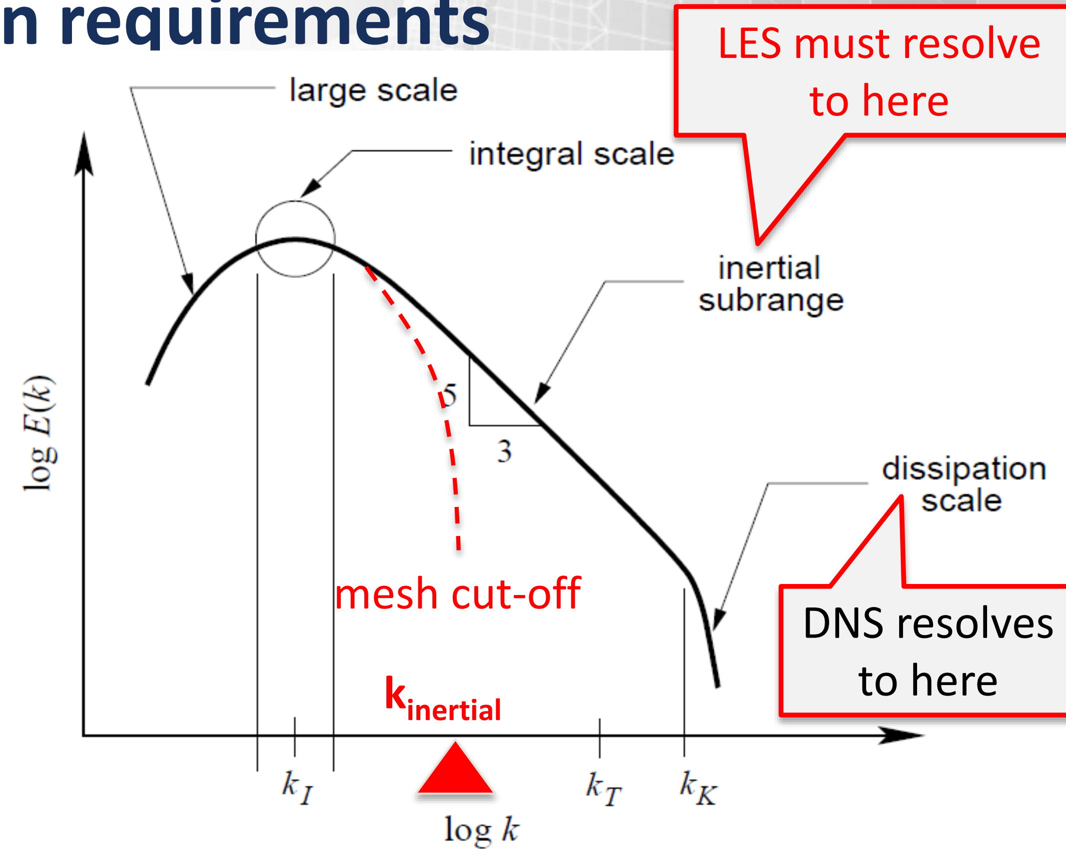



| Order | DOF/cell | PPW | Mesh size (B) | Relative size | DOF (B) | Floats/Δt/ mesh cell | Relative floats |
|---|---|---|---|---|---|---|---|
| P1 | 8 | 75 | 421. | 1 | 16840. | $3.6\text{x}10^3$ | 1 |
| P2 | 27 | 15 | 3.3 | 0.0078 | 445. | $6.9\text{x}10^3$ | 1.94 |
| P3 | 64 | 8 | 0.51 | 0.0012 | 163. | $14.7\text{x}10^3$ | 4.09 |

# Example: LES resolution requirements

- The Table shows the mesh sizes that would be needed to resolve to the target wavenumber
  $k_{inertial}$ ~ 3140.

- Also shown are the total Degrees of Freedom (DOF) and an estimate of the floating point operations needed (from our own code – *HOTnewt*)




| Order | DOF/cell | PPW | Mesh size (B) | Relative size | DOF (B) | Floats/Δt/ mesh cell | Relative floats |
|---|---|---|---|---|---|---|---|
| P1 | 8 | 75 | 421. | 1 | 16840. | $3.6\text{x}10^3$ | 1 |
| P2 | 27 | 15 | 3.3 | 0.0078 | 445. | $6.9\text{x}10^3$ | 1.94 |
| P3 | 64 | 8 | 0.51 | 0.0012 | 163. | $14.7\text{x}10^3$ | 4.09 |

# Example: relative cost

- The difference in mesh size is astonishing – but of course higher order methods consume more floats per cell to achieve the higher accuracy – so if we define the ***Relative Cost = relative mesh size x relative floats*** we see:

| Order | Relative Cost |
|---|---|
| P1 | **1** |
| P2 | 0.015 = 1/**66.2** |
| P3 | 0.0049 = 1/**203**. |

- The potential reduced Relative Cost – which translates directly into reduced computer energy requirements – derived from higher order methods is astonishing – and, as we'll see, the key to unlocking their potential is being able to develop a sufficiently coarse, higher order mesh
- ***This has strong implications for the energy cost of the simulation - roughly - 1 Gflop needs 1 Amp !***

CAMBRIDGE FLOW SOLUTIONS

# Example: relative cost

- The difference in mesh size is astonishing – but of course higher order methods consume more floats per cell to achieve the higher accuracy – so if we define the ***Relative Cost = relative mesh size x relative floats*** we see:

| Order | Relative Cost |
|---|---|
| P1 | **1** |
| P2 | 0.015 = 1/**66.2** |
| P3 | 0.0049 = 1/**203**. |

- The potential reduced Relative Cost – which translates directly into reduced computer energy requirements – derived from higher order methods is astonishing – and, as we'll see, the key to unlocking their potential is being able to develop a sufficiently coarse, higher order mesh
- ***This has strong implications for the energy cost of the simulation - roughly - 1 Gflop needs 1 Amp !***

CAMBRIDGE FLOW SOLUTIONS

# Higher Order Simulation Methods

- So, given the apparent overwhelming advantages of higher order methods, *why are they not in use in industry*?

- The key issues are implementing these methods on real-world, realistic meshes which usually are hybrid, consisting of hexahedra, tetrahedral, prisms & pyramids, since this is the only meshing style which can be automated and can be applied to complex industrial geometries

AND integrating the high order method within a process chain giving access at one end to higher order geometry – for higher order meshes – and then at the other end higher order post-processing

- Hence: the *BOXER* Environment & *HOTnewt*
  - see AIAA 2015-0833, AIAA 2016-0555, AIAA 2017-0742,…

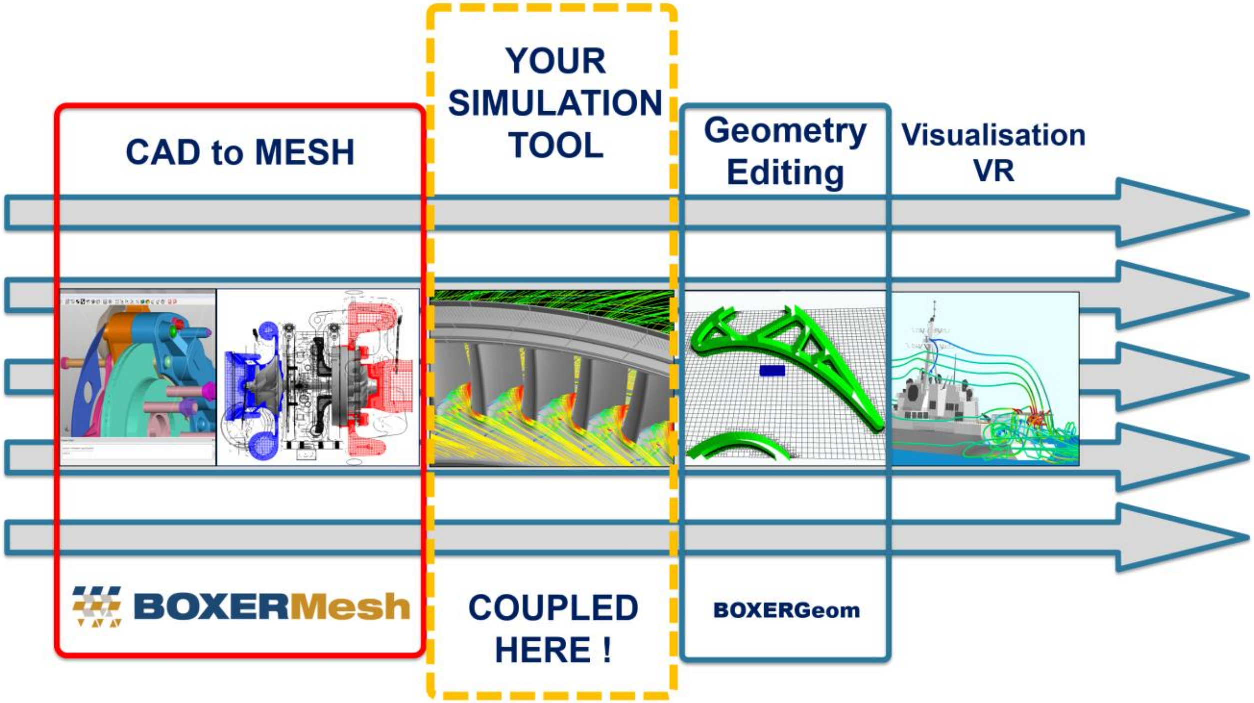
CAMBRIDGE FLOW SOLUTIONS

# Overview

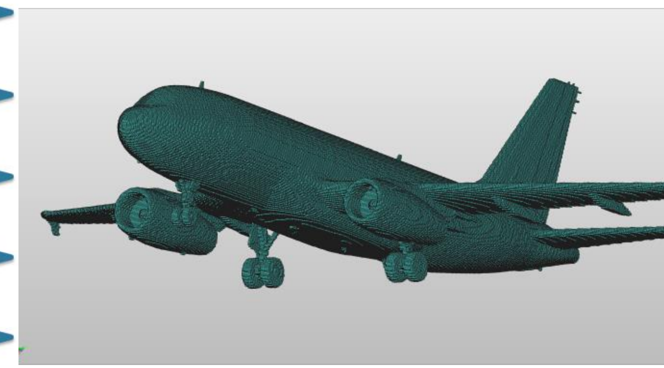
- Paper objective
- Why higher order ?
- **The BOXER Environment & LES/*HOTnewt***
- Turbomachinery & aerospace examples
- Computer resources
- Post-processing
- Summary

# BOXER Environment

- Supports an **End-to-End Parallel** Simulation Environment
  - Built on **Digital Geometry**
  - Fully scalable
  - Written with the flexibility to leverage future HPC & distributed parallel computing
- Very closely aligned with NASA's 2030 vision for revolutionary CFD

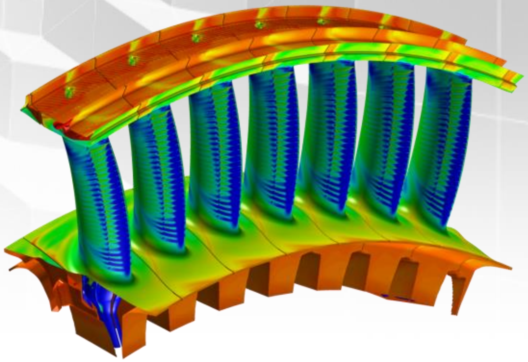



- **Digital geometry**

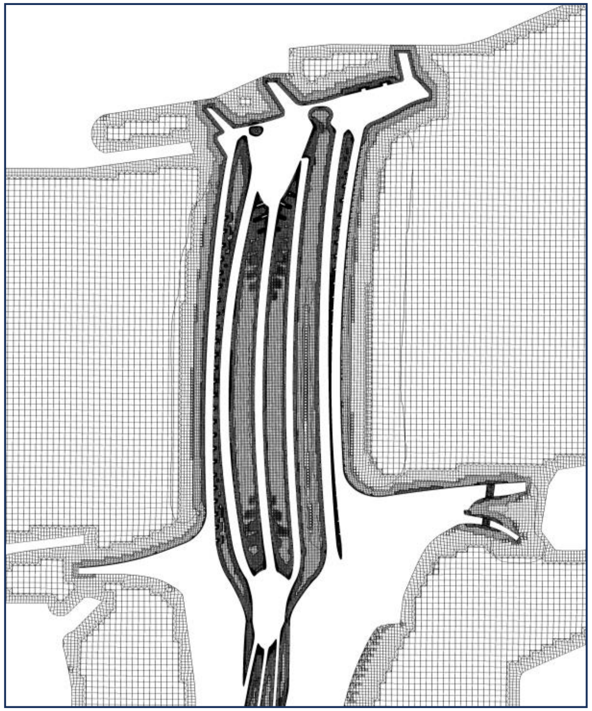


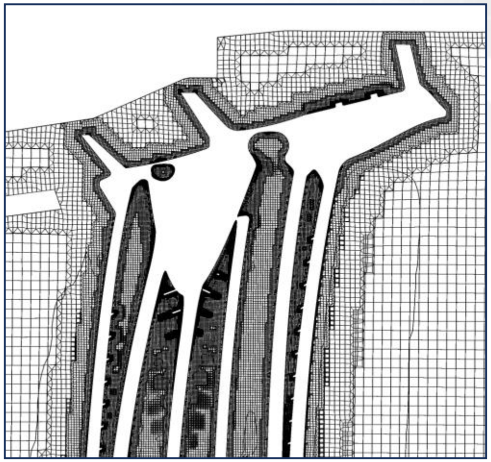
# BOXERMesh – Complex geometry

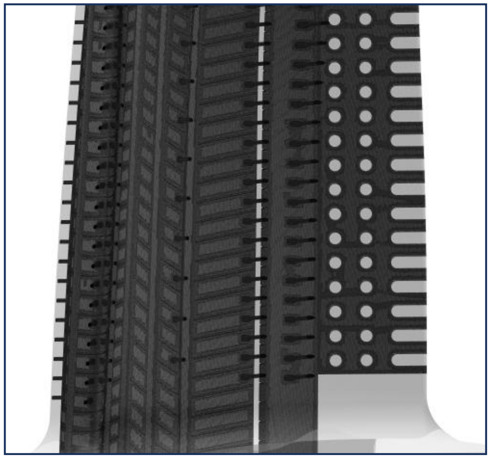
- An HP turbine rotor including cooling air system, shroud and under-hub

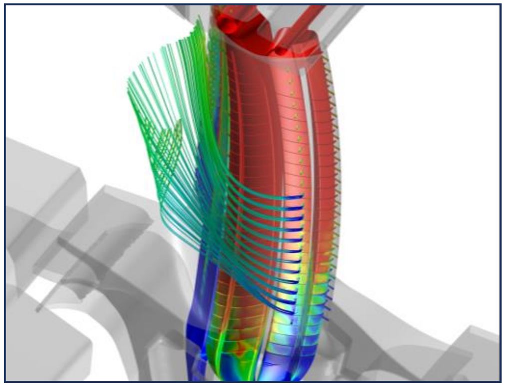


Mesh of the main gas path and cooling passages



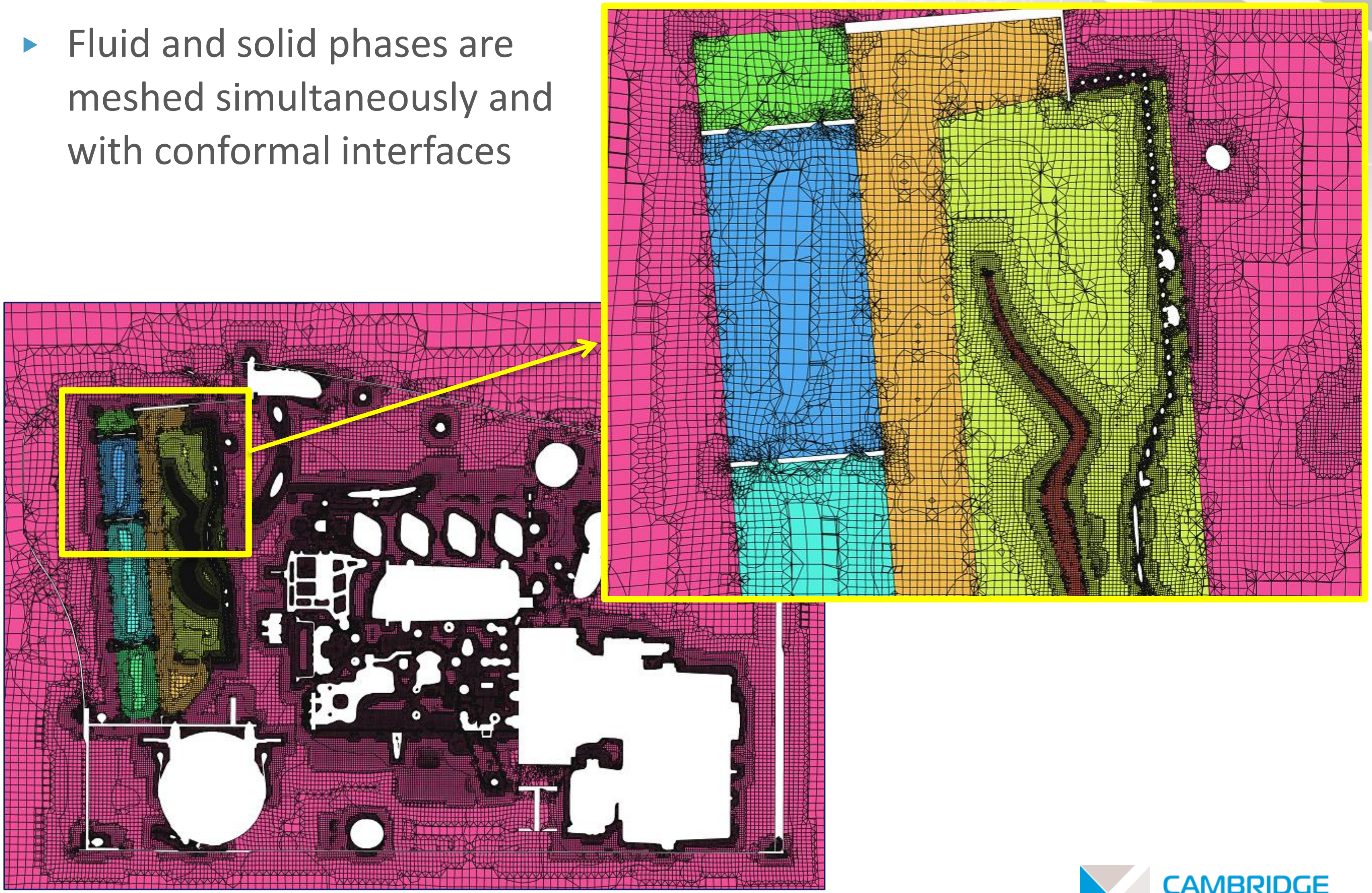






# BOXERMesh – Multi-region

- Fluid and solid phases are meshed simultaneously and with conformal interfaces

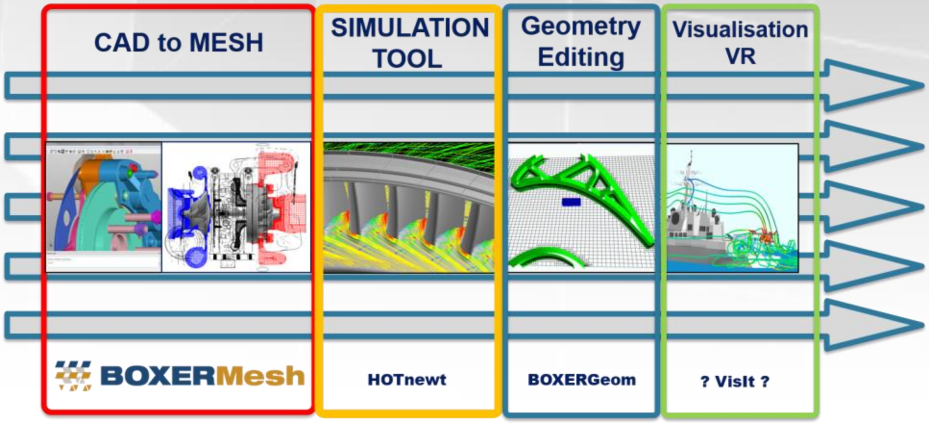


CAMBRIDGE
FLOW SOLUTIONS
Proprietary and customised CFD

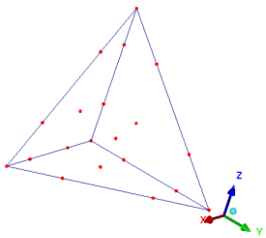
# CFD SOLVERS

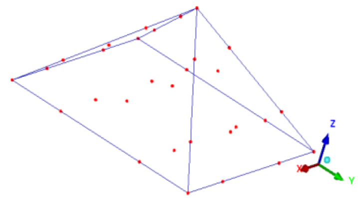
## HOTnewt

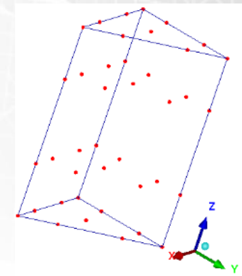
Fast, efficient higher-order LES

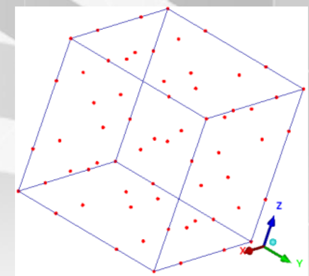


# HOTnewt - next generation LES flow solver







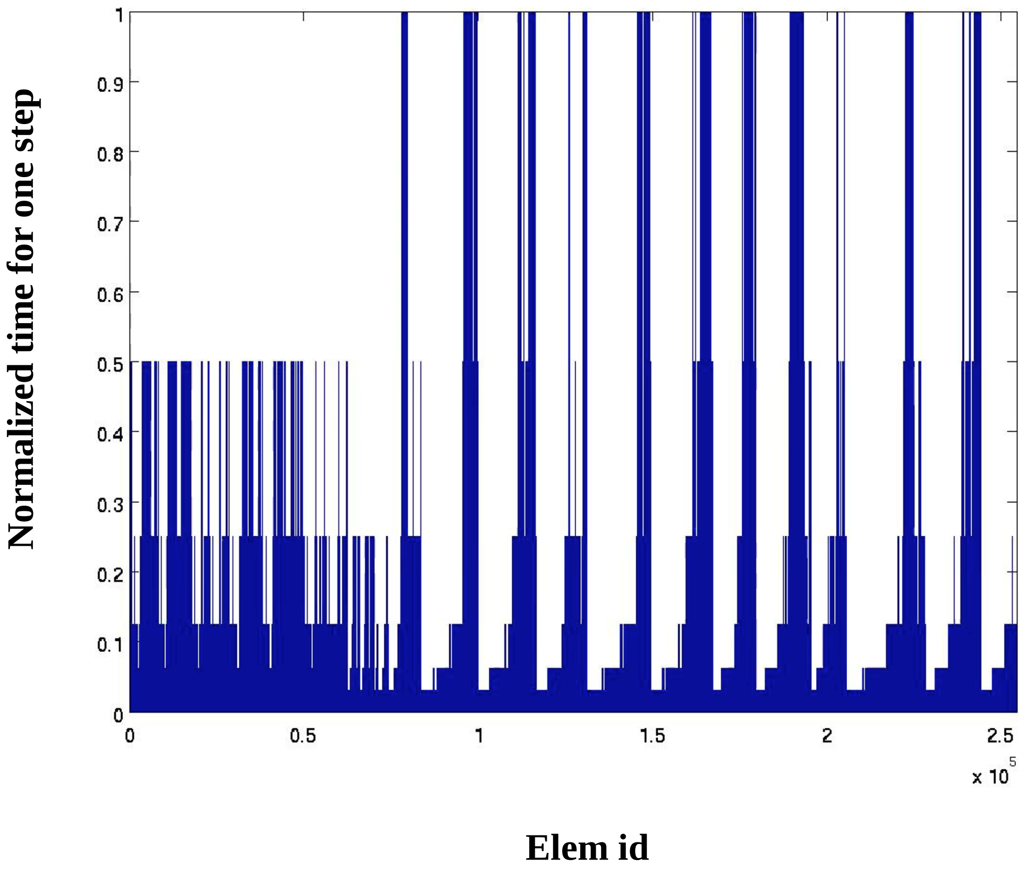


- Practical: runs on full hybrid unstructured (Boxer) meshes
- Efficient: Explicit approach using MPI+OpenMP – low memory footprint – ideal for **many-core / coprocessor systems (like the Intel PHI…..)**
- Fast: time-accurate local time-stepping giving potential speed-up of 10 to 100x relative to conventional (uniform time step) algorithms – **novel STEFR scheme**
  - Local space-time-extension for time marching, local Flux Reconstruction for space (based on Huynh [2007])
  - Low-memory: as low as 32Gb / Mcell in 3$^{rd}$ order, wall resolved
  - Wall-resolved or wall-modelled (implicit sub-grid model based on Park & Moin [2014])
  - Quadrature-free, differential form, no mass matrix
  - Up to 4$^{th}$ order accurate in space

CAMBRIDGE FLOW SOLUTIONS

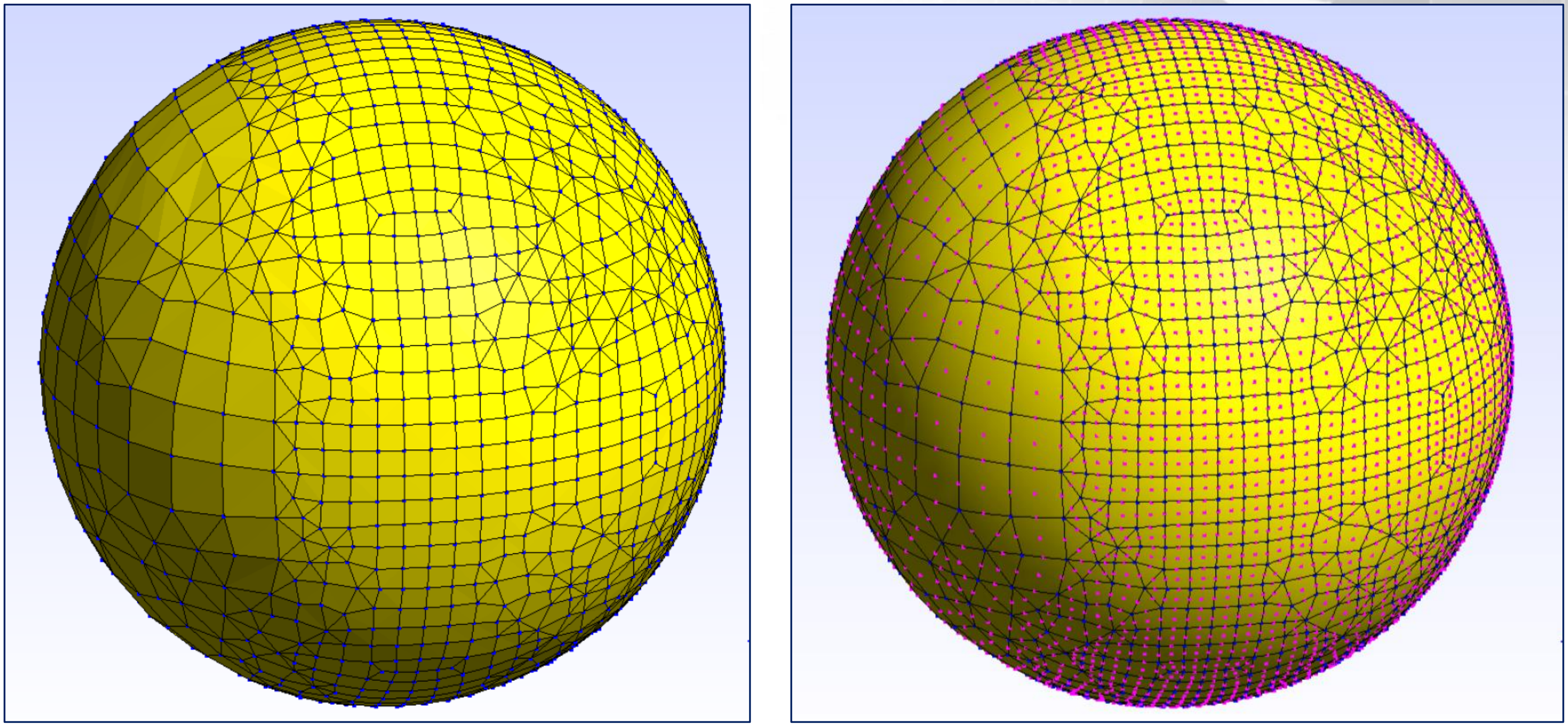
# Space Time Extension of FR (STEFR)

- We use *local* time stepping within the framework of a special time integration which *preserves time accuracy*
  - For each time step in the larger cells many steps are taken in the smaller cells
  - The main challenge is maintaining an efficient parallel load balance
- This has the potential to be 10~100 times faster on hybrid meshes for realistic industrial-class problems than classic, explicit time-marching (eg. R-K)




CAMBRIDGE FLOW SOLUTIONS

# HOTnewt - Flow past a sphere

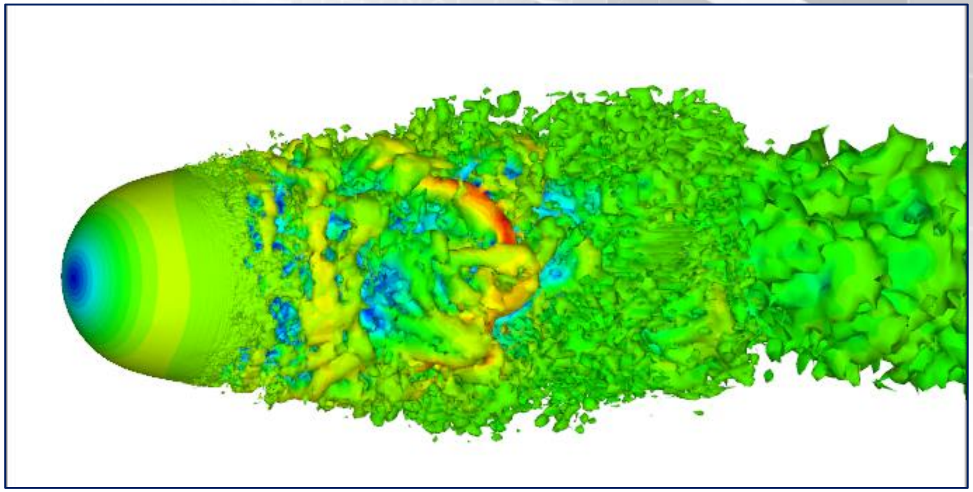


- Benefit of higher-order geometry: higher fidelity representation at equal or lower cell count

CAMBRIDGE FLOW SOLUTIONS

# HOTnewt – Higher order improves accuracy for same cost

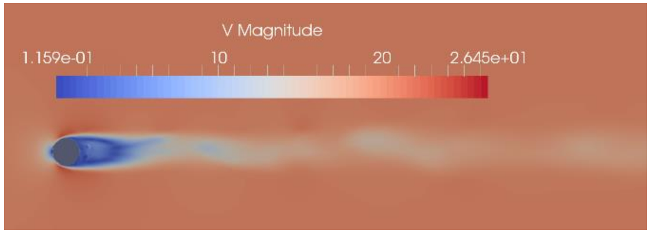
- Flow past a **sphere** at Re=300 & Re=10,000
- Calculated $2^{nd}$, $3^{rd}$ and $4^{th}$ order runs of similar computational cost
  - Increased order *reduces* mesh cell count & DoFs



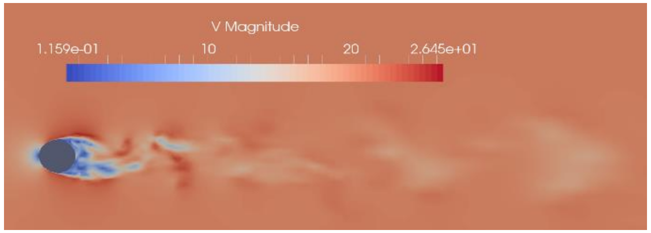
Instantaneous Q-criterion – Re = 10,000 $3^{rd}$ order

Instantaneous velocity magnitude

**Re = 300**



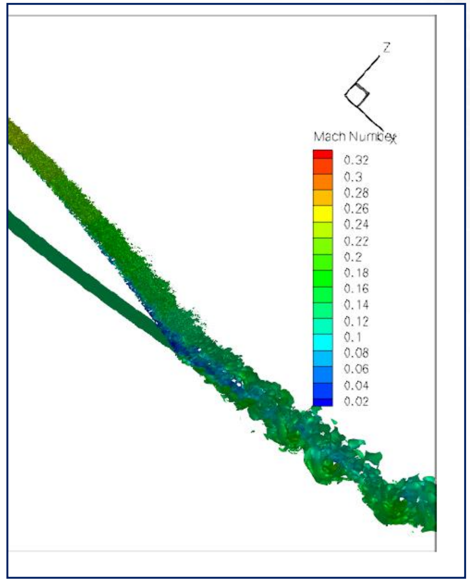

**Re = 10,000**

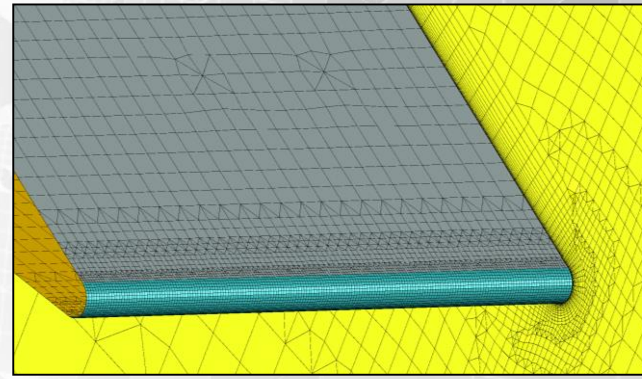


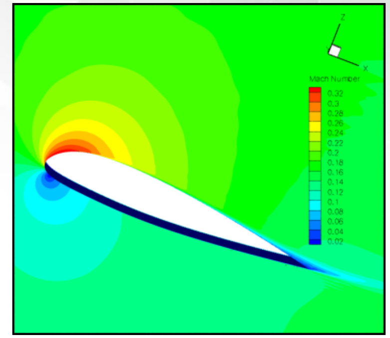

| Reynolds number | Order of accuracy | Ncells | Ndofs | Cd | Cd (combined calc & expt) |
|---|---|---|---|---|---|
| 300 | $2^{nd}$ | 419250 | 13.9M | 0.676 | |
| 300 | $3^{rd}$ | 155286 | 15.2M | 0.664 | |
| 300 | $4^{th}$ | 52447 | 10.7M | 0.658 | 0.657 |
| 10000 | $2^{nd}$ | 3783863 | 135.3M | 0.448 | |
| 10000 | $3^{rd}$ | 735883 | 79.2M | 0.439 | |
| 10000 | $4^{th}$ | 164013 | 37.7M | 0.429 | 0.416 |

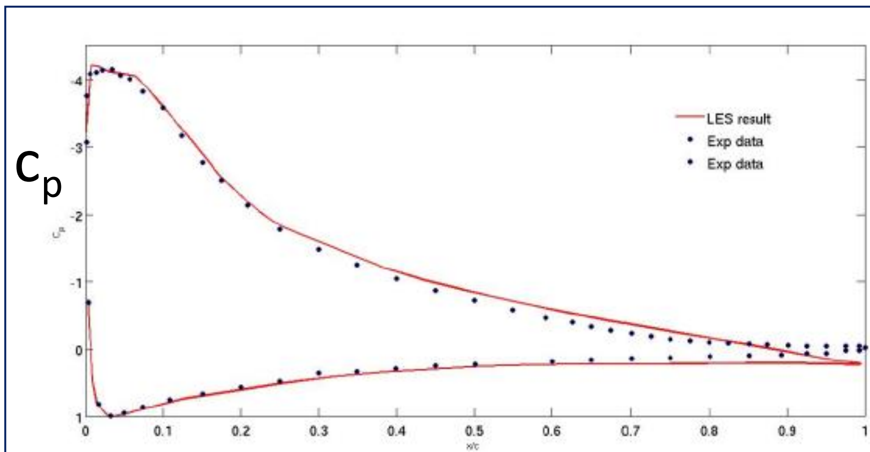
# HOTnewt – LES for an airfoil

- Standard airfoil
- Re = $2.1 \times 10^6$ and $\alpha$=13.3°; span = 3% of chord
- $2^{nd}$ order with local $3^{rd}$ order p-refinement – 15.2M DoF
- **wall-resolved** $Y^+ \sim 1.3$

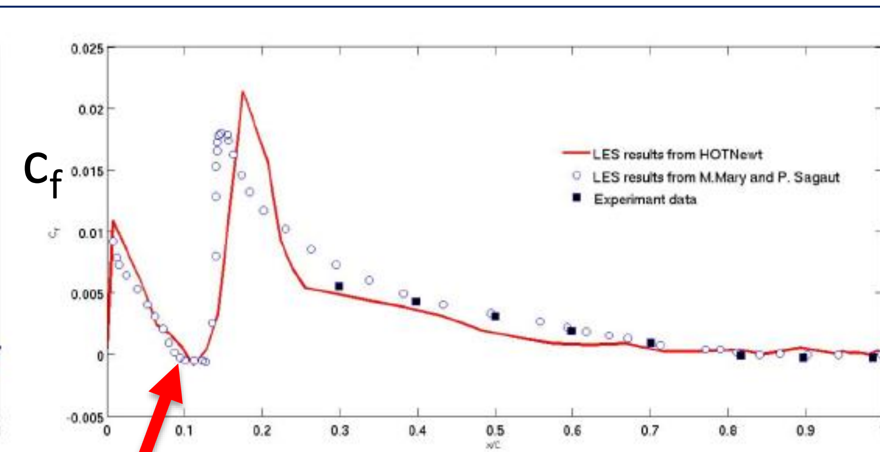





Mean Pressure coefficient




Mean Skin-friction coefficient

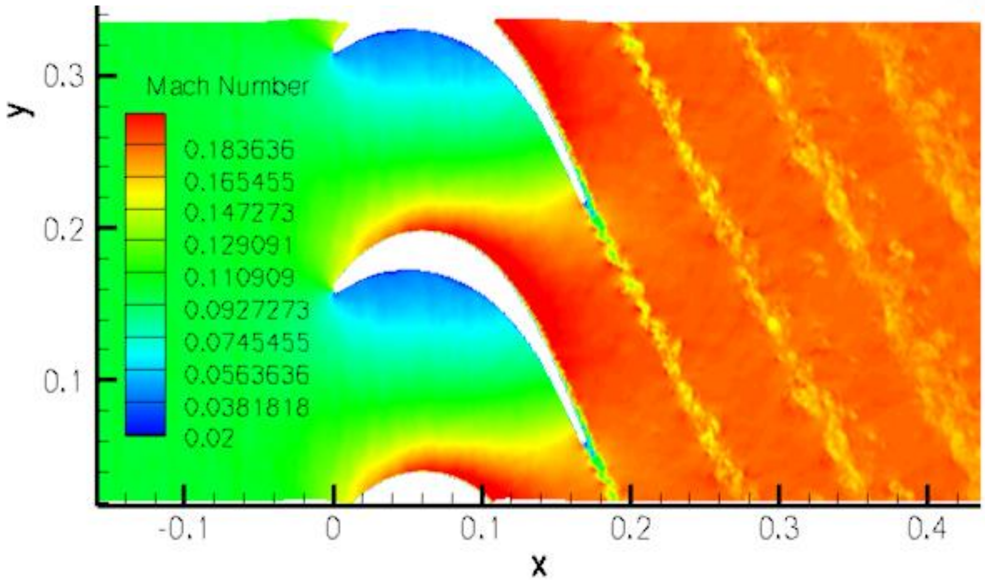
**Wall-resolved HOTnewt LES predicts transition**

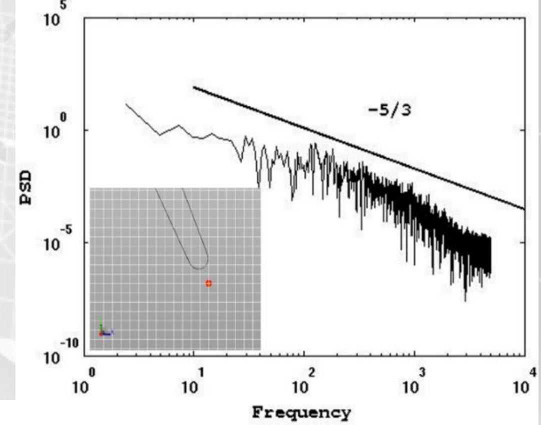
CAMBRIDGE FLOW SOLUTIONS

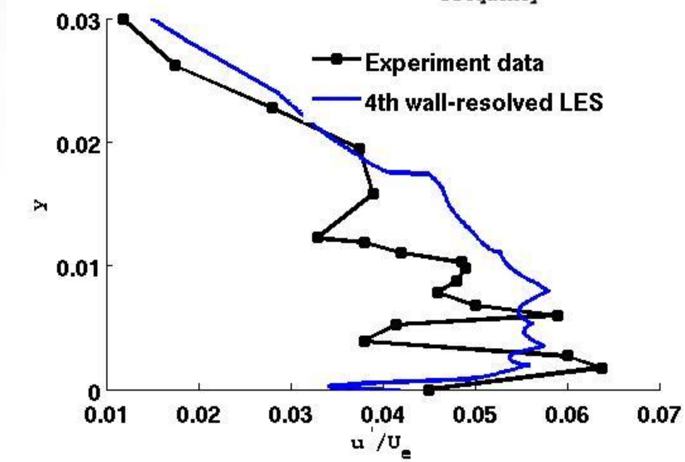
# Overview

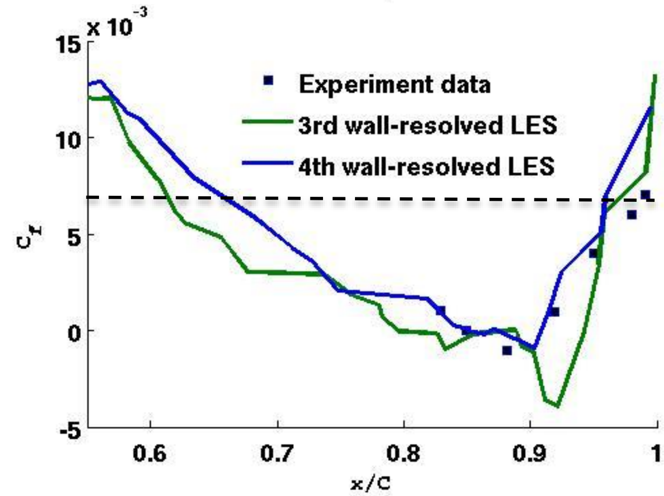
- Paper objective
- Why higher order ?
- The BOXER Environment & LES/*HOTnewt*
- **Turbomachinery & aerospace examples**
- Computer resources
- Post-processing
- Summary

# HOTnewt – Low pressure turbine: T106



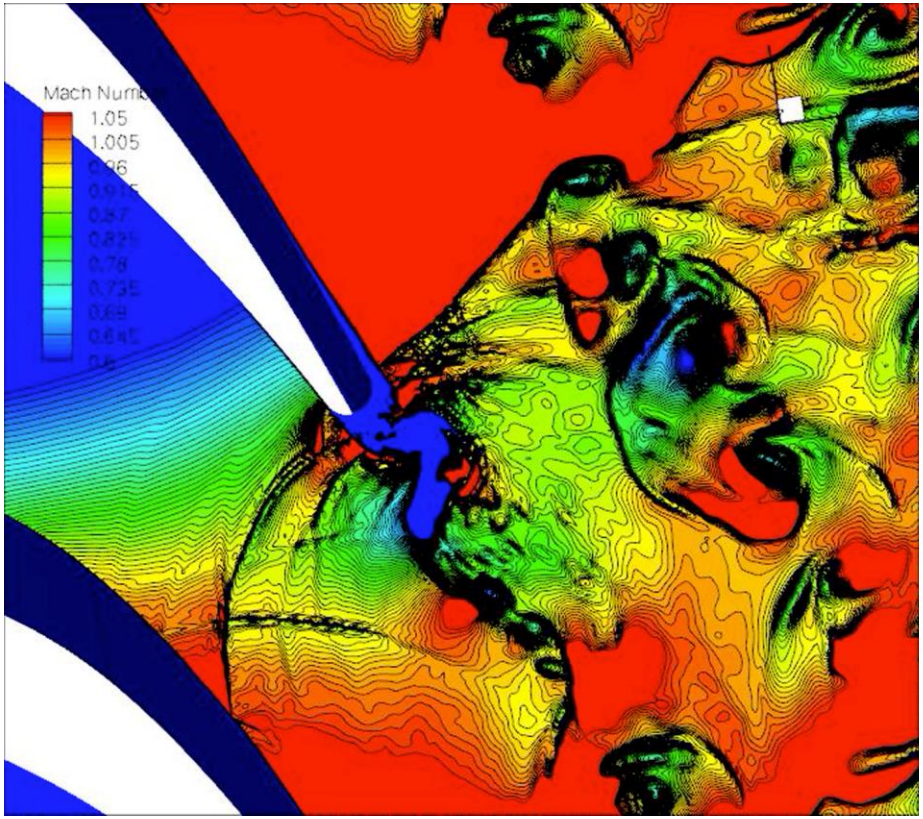



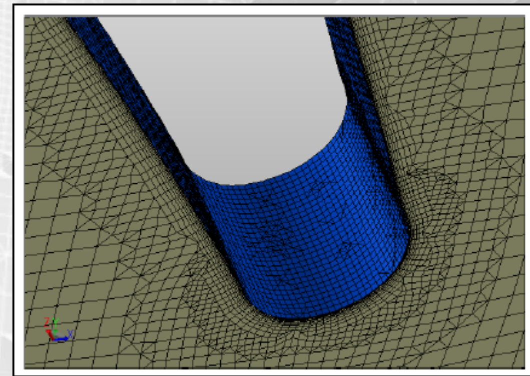



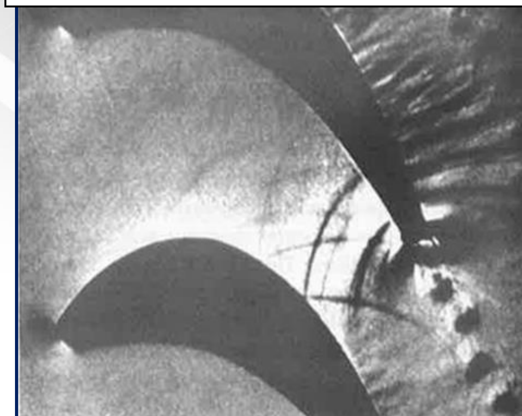

- Tested at the Whittle: suction side boundary layer transition & separation bubble
- $\mathrm{Re}=1.1 \times 10^5$; Inlet Mach is 0.1
- Spanwise extent of the domain $L_z = 0.075C$
- STEFR speed-up ≈ 8.1 rel. to conventional, uniform time-stepping

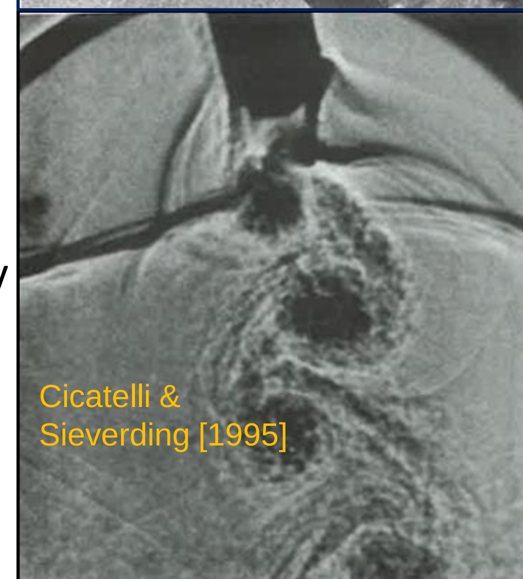
- **4th order wall-resolved LES, 110k cells**

# HOTnewt – Transonic turbine: VKI





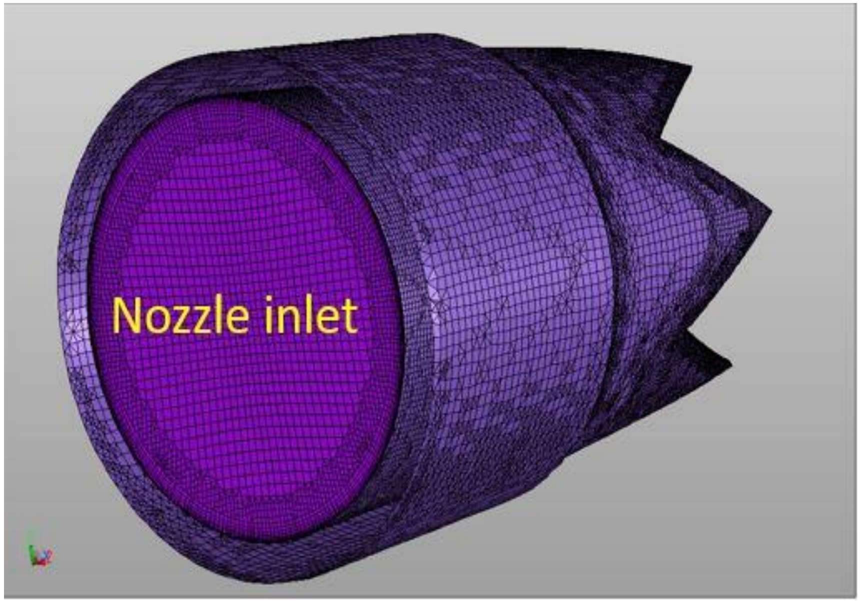


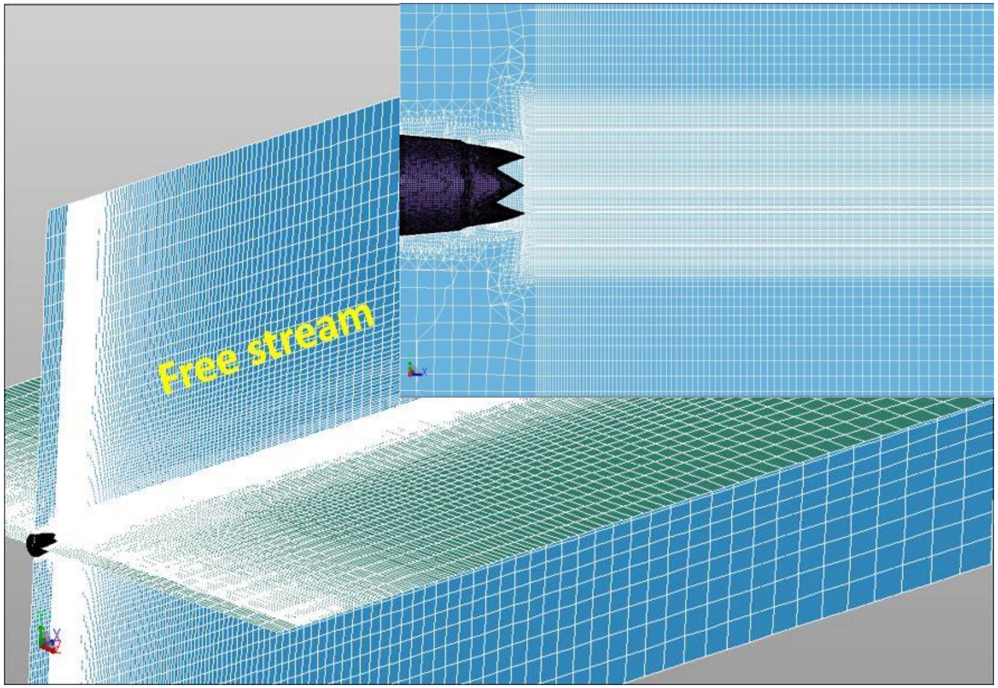

- Tested at the VKI: reflected shock at the suction side boundary layer interaction; shock-trailing edge vortex wake interactions
- Re=$8.5 \times 10^5$, **3rd order**, STEFR speed-up of ~8.1 compared to conventional methods
- **Wall-resolved LES, 107k cells**, with 137.3M DOFs

# NASA Acoustic Reference Nozzle

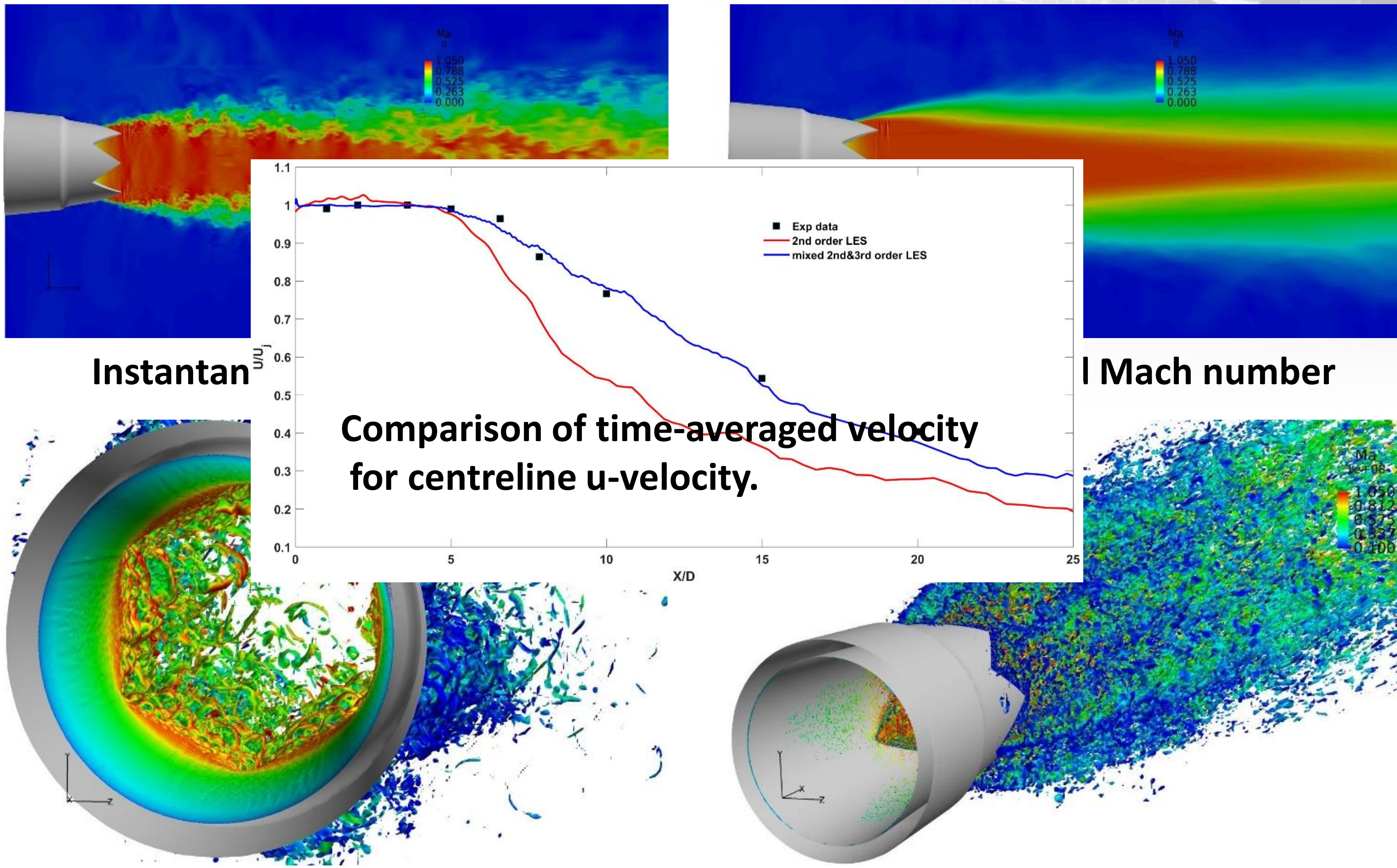
| Geometry | Nozzle inlet total pressure $p_0$(Pa) | Nozzle inlet total temperature $T_0$(K) | Free stream static pressure $p_\infty$(Pa) | Free stream temperature $T_\infty$(K) |
|---|---|---|---|---|
| SMC001 nozzle | $1.78 \times 10^5$ | 286.4 | $9.7 \times 10^4$ | 280.2 |

| Case ID | Order of accuracy | Number of cells | Number of DOFs | Speed Up Ratio | Number of nodes on our cluster | Equivalent number of CPU cores | Memory consumption(Gb) | Wall-clock time for $1T_p$ (hours) |
|---|---|---|---|---|---|---|---|---|
| SMC001_1 | $2^{nd}$ | 9.04M | 338.4M | 16.3 | 1 (8 PHI cards) | 80 | 167 | 5.09 |
| SMC001_2 | mixed $2^{nd}$ /$3^{rd}$ | 7.66M | 671.1M | 29.2 | 4 (3 PHI cards each) | 160 | 317.4 | 9.72 |



# NASA Acoustic Reference Nozzle



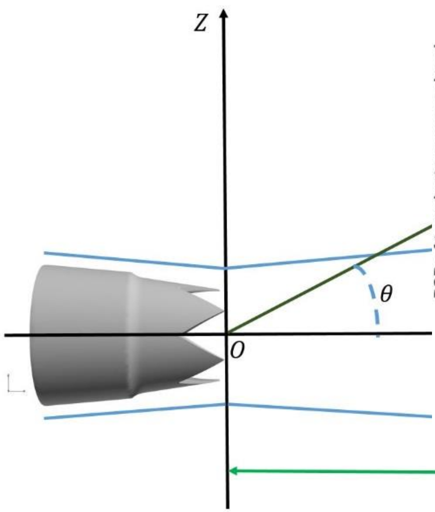

Instantan… …l Mach number

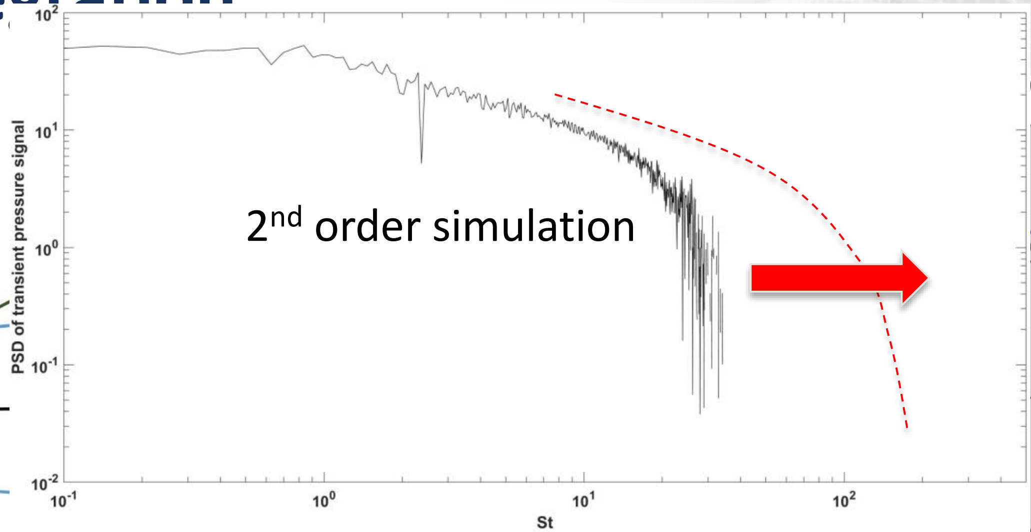
**Comparison of time-averaged velocity for centreline u-velocity.**

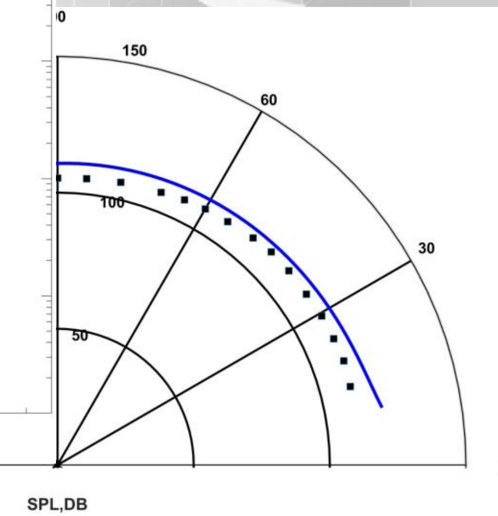
**Snapshots of instantaneous Q-criterion illustrating the jet structure, coloured by Mach number**

CAMBRIDGE FLOW SOLUTIONS

# Acoustic post-processor: *ffowcs* Williams-Hawkings (FWH) integration

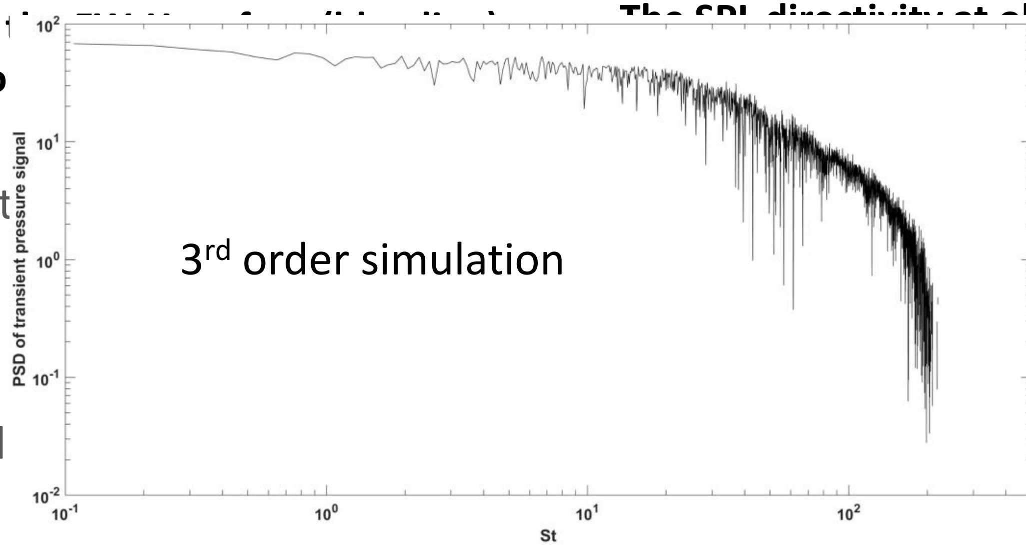



**Representation of t**
**and the far-field o**

**The SPL directivity at observation points with**
**the present simulation**
**ment**

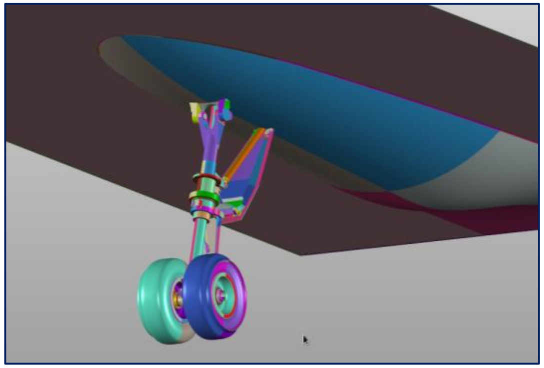



- far-field acoust … int
- far-field overal

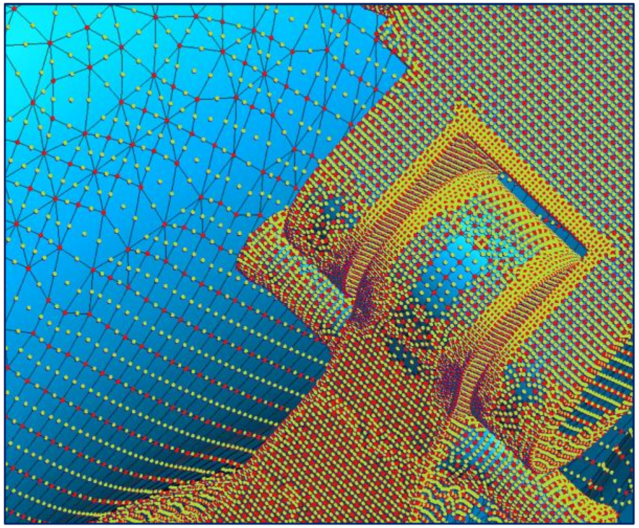
$dS$

# HOTNewt – Landing gear aero-acoustic case

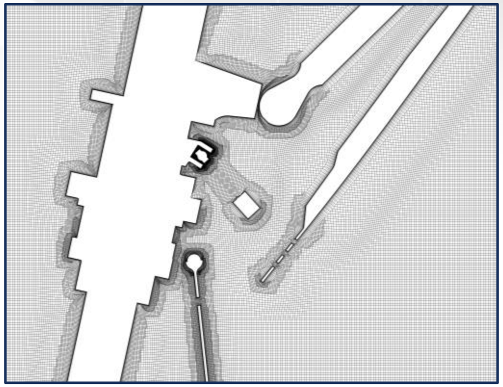
- BANC II – Landing gear aero-acoustic test case
  - **3rd order, wall-modelled - 11.1M cells** - 862 M DoFs
  - 301Gb memory running on 2,736 cores on "*Ulysses*", our new CFS **12kW Intel PHI cluster**
  - Complete simulation in ~1 week
  - STEFR speed up is ~35 x faster than uniform time-step LES

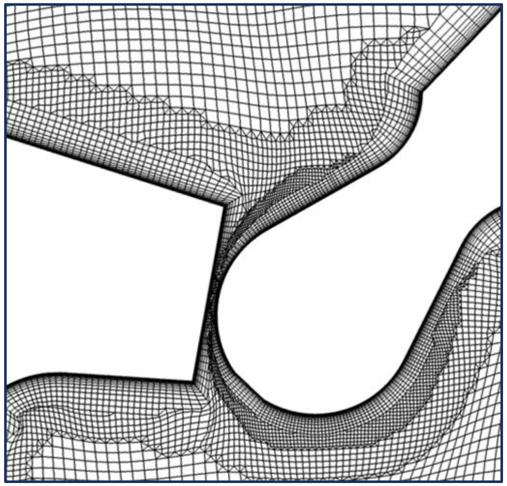


Landing gear geometry

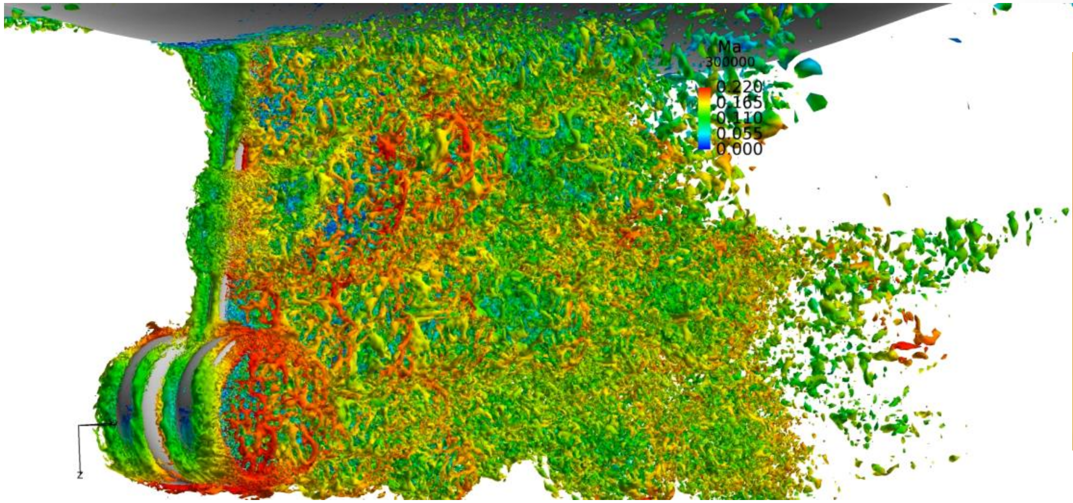


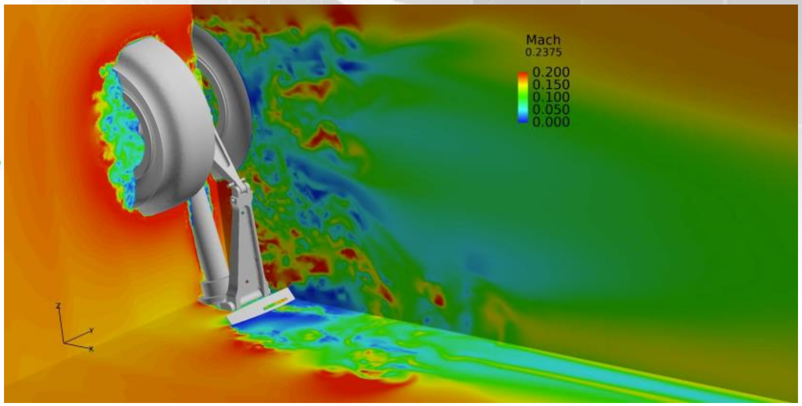
Detail of P2 mesh





# HOTNewt – Landing gear aero-acoustic case




Mach contours ($M_\infty$ = 0.2375)

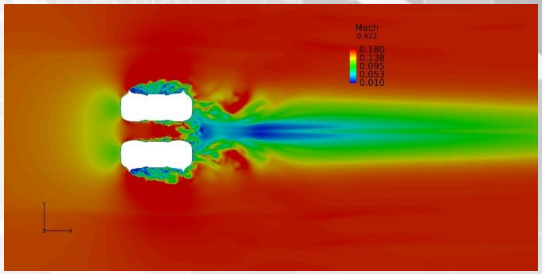
Instantaneous Q= 300,000

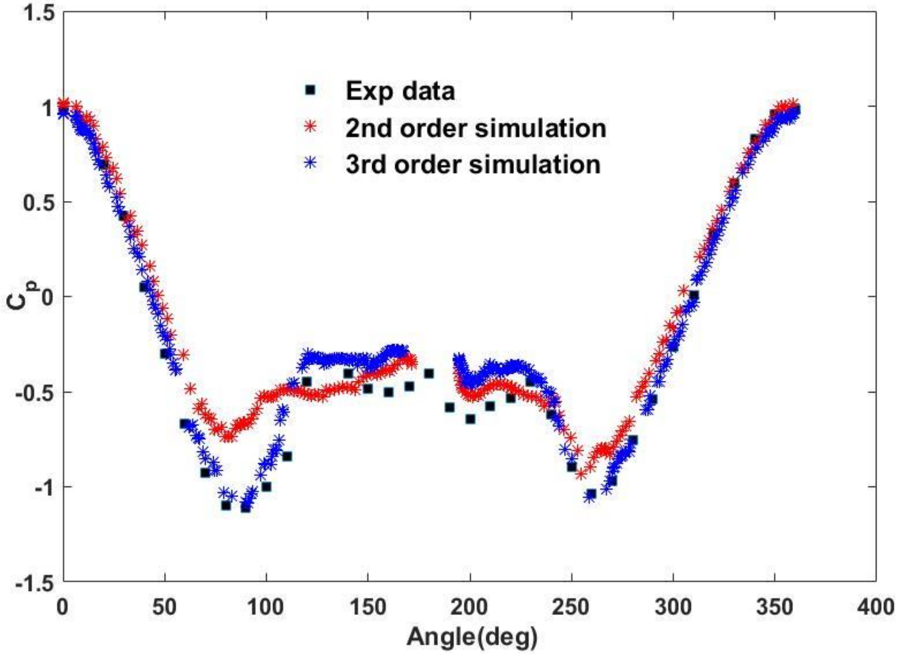
- Wall-modelled mesh: 3rd order
- 11.1M cells, 862M DOFs
- 18.7 wall-clock hours for one flow-past time, $\Delta T_p$ , based on strut size and run on “*Ulysses*” our **12kW Intel PHI system**
- Complete simulation in ~ 1 week

CAMBRIDGE FLOW SOLUTIONS

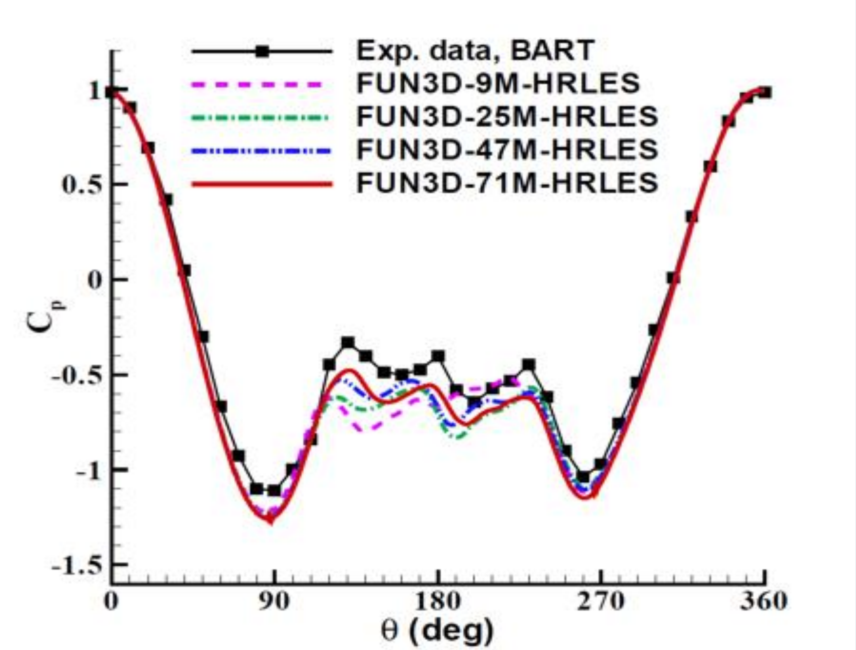
# HOTNewt – Landing gear aero-acoustic case

- 3rd order result is better, with no over-prediction relative to data
- Favourable comparison with NASA/FUN3d run on much finer meshes




HOTnewt 2nd & 3rd order $C_p$ distribution around port wheel




(a) Grid sensitivity for HRLES model

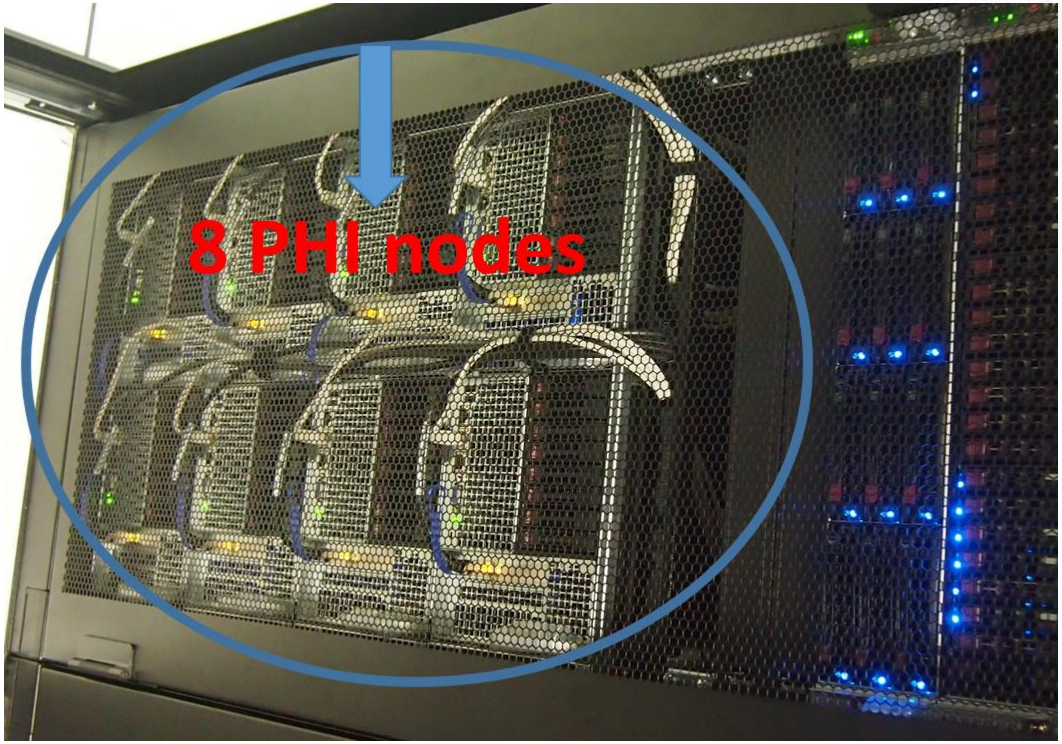
**Surface pressure distributions on port wheel**

CAMBRIDGE FLOW SOLUTIONS

# Overview

- Paper objective
- Why higher order ?
- The BOXER Environment & LES/*HOTnewt*
- Turbomachinery & aerospace examples
- **Computer resources**
- Post-processing
- Summary

# CFS Intel Phi Cluster - our *12kW Intel PHI system*




**"Ulysses"**

**Architecture: 16 Intel Xeon E5-2692 8C, 48 Xeon PHI 31S1P cards.**

**Power: 12 KW**

**Memory: 1TB on CPUs and 384GB on PHI cards**

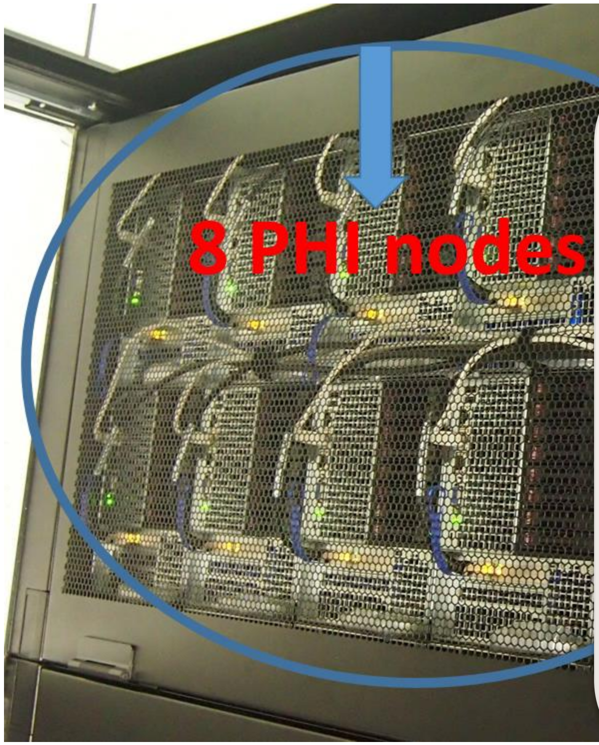
**Speed: ~30 TFLOPS**

**Cost: ~$75k = $0.075 milion**

- Built using an Innovate UK SMART award, "Ulysses" is equivalent to about 600-800 conventional CPU cores depending on loading but a fraction of the cost compared to CPU cluster
- **Equivalent to less than 1/10 the cost of a CPU-core-hour (Jaeggi [2016])**
- Ideal for intensive simulations, low hardware & running cost

CAMBRIDGE FLOW SOLUTIONS

# CFS Intel Phi Cluster - our *12kW Intel PHI system*




**"Ulysses"**

**Architecture: 16 Intel Xeon E5-2692 8C, 48 Xeon PHI 31S1P cards.**

**Power: 12 KW**

**Memory: 1TB on CPUs and 384GB on PHI cards**

**Speed: ~30 TFLOPS**

**Cost: ~$75k = $0.075 milion**

***VS***

**Top 1 HPC: Tian-he 2**

**Architecture: 32,000 Intel Xeon E5-2692 12C, 48,000 Xeon PHI 31S1P cards.**

**Power: 17.6 MW (24 MW with cooling)**

**Memory: 1000TB on CPUs and 375TB on PHI cards**

**Speed: 33.86 PFLOPS**

**Storage: 12.4PB**

**Cost: US$390 million**

- "Ulysses" is equivalent to about to ~700 conventional CPU cores but consumes 12kW compared to ~192kW for ~700 CPU cores within Tian-he 2

  **factor x16 less electricity and x57 cheaper to buy !**

- HOTnewt is highly optimised to run on this type of architecture

CAMBRIDGE FLOW SOLUTIONS

# Computer resource comparison: BANC case

- It is difficult to find comparative computer resource data in the open literature – and comparisons are made doubly difficult by the different mesh sizes and algorithms
- Khorrami [2015] does give data for the ONERA code CEDRE [2011] for the BANC test case
  - CEDRE was run as a second order solver with a 70M cell mesh on 480 conventional cpu cores and needed 1.44 hours per $\Delta t_{P.}$
- The present *HOTnewt* simulations were
  - run 3rd order with 11.1M cells (862 M DoFs) - wall-clock 18.7 hours of our 12kW Intel PHI system were needed for one flow passing period $\Delta t_P$
- Hence, the **raw comparison** is:
  - ONERA/CEDRE: 480cpu_**1.44** (cores_wall clock hours/ $\Delta t_{P.}$ )
  - HOTnewt: 48phi_**18.7** (cores_wall clock hours/ $\Delta t_{P.}$ )
- BUT how do we allow for the different hardware, mesh size & order ?

CAMBRIDGE FLOW SOLUTIONS

# Computer resource comparison: BANC case

- **Hardware**:
  - our Intel PHI cluster benchmarks to be equivalent to ~700 conventional cpu cores so we can simply scale the *HOTnewt* simulations to 480 conventional cores
- Hence, the comparison **scaled for hardware** is:
  - ONERA/CEDRE: 480_**1.44** (cores_wall clock hours/ $\Delta t_{P.}$ )
  - HOTnewt: 480_**27.3** (cores_wall clock hours/ $\Delta t_{P.}$ ) [scaled hardware]
- On the face of it *HOTnewt* is therefore much slower than CEDRE…
- …BUT the mesh size & order are different…

CAMBRIDGE FLOW SOLUTIONS

# Computer resource comparison: BANC case

- **Mesh size & order (1)**:
  - 70M cells in a second order solver is equivalent to $(15/75)^3 \times 70M = 0.55M$ cells in a third order solver using the PPW data from earlier
  - Scaling the computer work between meshes is partly the ratio of mesh sizes (this is simply the basic floating point work) but also the (ratio of mesh sizes)$^{1/3}$ to approximately scale the time step change assumed limited by a CFL number criterion

- Hence, the comparison **scaled for hardware & mesh/order** is:
  - ONERA/CEDRE: 480_**1.44** (cores_wall clock hours/ $\Delta t_{P.}$ )
  - HOTnewt: 480_**0.49** (cores_wall clock hours/ $\Delta t_{P.}$ ) [scaled hardware, mesh size]

- This indicates *HOTnewt* would be a factor 2.9 ***faster*** than ONERA/CEDRE

CAMBRIDGE FLOW SOLUTIONS

# Computer resource comparison: BANC case

- **Mesh size & order (2)**:
  - An alternative approximate scaling is to try to go the other way – from 3rd order to 2nd order
  - The *HOTnewt* 11.1M cell 3rd order mesh is equivalent to a $(75/15)^3$x11.1M =1,423M cell 2nd order mesh !
  - The run time for this would scale as (mesh size)$^{4/3}$ as described above

- Hence, the comparison **scaled for hardware & mesh/order** is:
  - "2nd order solver": 480_**79.8** (cores_wall clock hours/ $\Delta t_{P.}$ ) [scaled mesh size]
  - HOTnewt: 480_**27.3** (cores_wall clock hours/ $\Delta t_{P.}$ ) [scaled hardware]

- This again indicates *HOTnewt* would be a factor 2.9 ***faster*** than a "2nd order solver"
  - when combined with the factor ~10 reduced energy consumption of our Intel PHI system ***HOTnewt* LES would appear to be around a factor ~29 cheaper than other LES methods for comparable resolution**.

CAMBRIDGE FLOW SOLUTIONS

# Overview

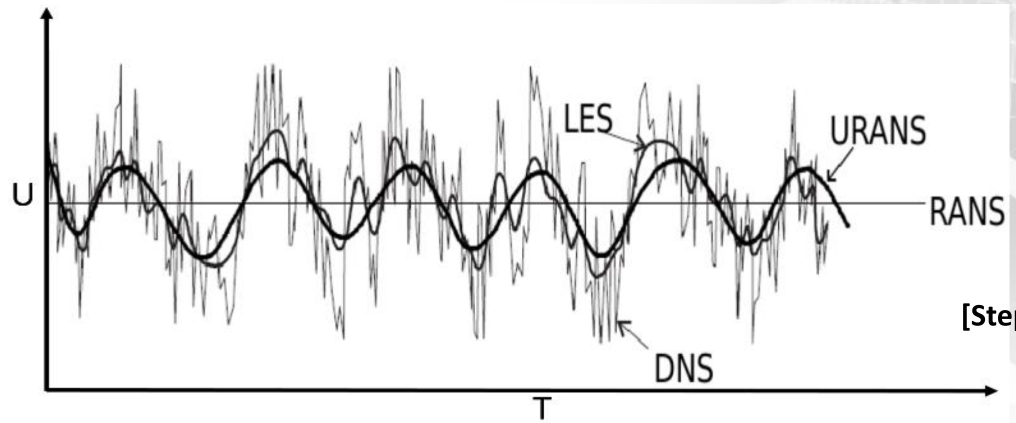
- Paper objective
- Why higher order ?
- The BOXER Environment & LES/*HOTnewt*
- Turbomachinery & aerospace examples
- Computer resources
- **Post-processing**
- Summary

# Post-processing & “Big Data”

- The big challenge now is post-processing – getting useful, sensible engineering data out of these large simulstions
- Certainly post-processing will need to be implemented in parallel, and close-coupled into to the solver parallel data structures
  - But there are other approaches which may well contribute
- The main flow information (low frequency, most energy-containing - main industrially interesting statistics) could be efficiently reconstructed and analysed using low order orthogonal modal modes, which are remarkably fast and low memory
- *On-the-fly* POD data-extraction and analysis during the LES would allow these benefits to be achieved in huge, massively parallel Billon+ DOF simulations - suggesting that great efficiency in post-processing these “big data” problems is indeed achievable
  - Further, the connections between the low frequency part and high frequency part could be investigated further, case by case (high frequency part, turbulence, unsteadiness, …) to drive insight into improved low-order modelling like RANS…

CAMBRIDGE FLOW SOLUTIONS

# Hierarchical Proper Orthogonal Decomposition (HPOD)




[Stephen J. Lawson, 2007]

**Examples of time signals from DNS, LES, URANS and RANS simula**

Huge memory consumption for massive high fidelity unsteady results

- Review of Karhunen-Loeve Decomposition (KLD) met

$$U(x,t) = U_m + \sum_{i=i}^{N_t} c_i(t)\Phi_i(x)$$

- $N_t \leq N$ is the selected number of snapshots for reconstruction

$\boldsymbol{B} = (1/N)(\boldsymbol{A}^T\boldsymbol{A})$, $\boldsymbol{A}$ denote an $M \times N$ matrix of real data

$$\boldsymbol{B}v_i = \lambda_i v_i, \qquad for\ i \in [1,N], \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$$

$$\Phi_i = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{N} (v_i)_j U_j^h, \qquad for\ i \in [1,d]$$

CAMBRIDGE FLOW SOLUTIONS

# Multi-level orthogonal modal decomposition – *on-the-fly*

- The analysis is re-cast so that it can be performed on-the-fly closely integrated into the flow solver data structures
- Orthogonal modal solution

$$u_{i,j}^m = \int_E \phi_j^m U_i dE, \qquad j \in [1, N_{dofs}^m]$$

where $\int_E \phi_i^m \phi_j^m dE = v\boldsymbol{I}$

| Order of accuracy | 1st | 2nd | 3rd | 4th |
|---|---|---|---|---|
| Number of DOFs | 1 | 4 | 10 | 20 |
| Size ratio of $\boldsymbol{A}$ with same number of snapshots | 1 | **4** | **10** | **20** |

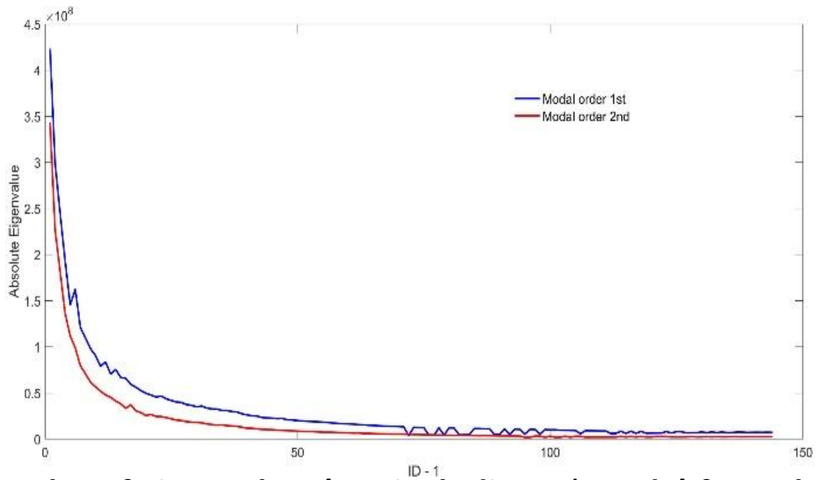
- The reconstruction of flow variables on a nodal point $\vec{p}$ on $i$ −th element is

$$u_i^n(\vec{p}) = \sum_j^{N_{dofs}^m} \phi_j^m(\vec{p}) u_{i,j}^m$$

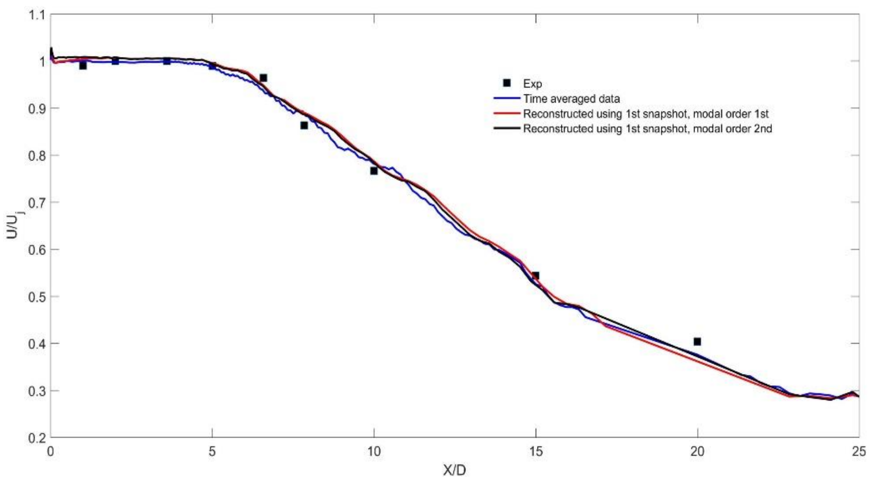
CAMBRIDGE FLOW SOLUTIONS

# HPOD analysis for SMC001 transonic nozzle case

| Job ID | Modal order | No. of variables per cell | Total DOFs | Size of instantaneous results (Gb) | No. of snapshots | Size of matrix $A$(single precision, Gb) | Actual memory consuming for HPOD(Gb) | Size of modal result for each snapshot(Gb) |
|---|---|---|---|---|---|---|---|---|
| SMC001-K0-Pressure | 1st | 1 | 7665463 | 1543.5 | 250 | 7.14 | 8.35 | 0.1591 |
| SMC001-K1-Pressure | 2nd | 1 | 30661852 | 1543.5 | 250 | 28.56 | 28.37 | 0.6364 |
| SMC001-K2-Pressure | 3rd | 1 | 76654630 | 308.71 | 100 | 28.56 | | 1.591 |
| SMC001-K0-Velocity | 1st | 3 | 22996389 | 1543.5 | 250 | 21.42 | 21.3 | 0.4773 |
| SMC001-K1-Velocity | 2st | 3 | 91985556 | 1543.5 | 250 | 85.68 | 85.2 | 1.9092 |
| SMC001-K2-Velocity | 3rd | 3 | $2.3 \times 10^8$ | 308.71 | 100 | 85.68 | | 4.773 |



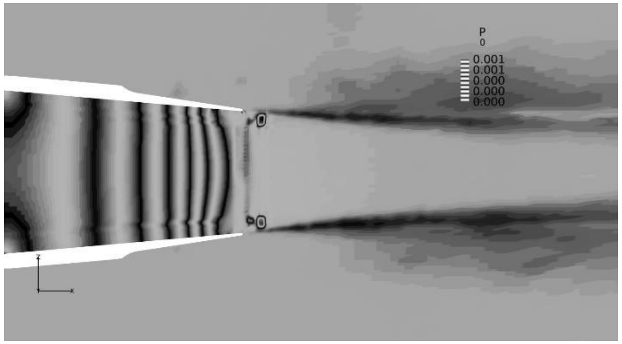

Plot of Eigenvalues(not including 1st mode) from the HPOD analysis



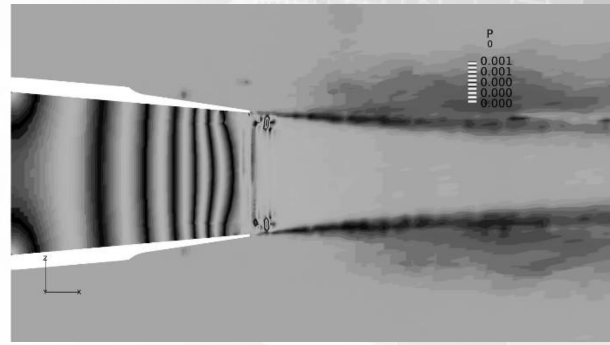

Comparison of reconstructed, time-averaged centreline u-velocity using the 1st POD mode

CAMBRIDGE FLOW SOLUTIONS

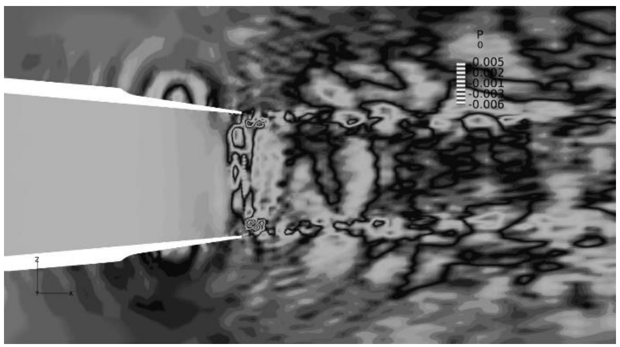
# POD modes of pressure field in the X-Z plane (Y=0)



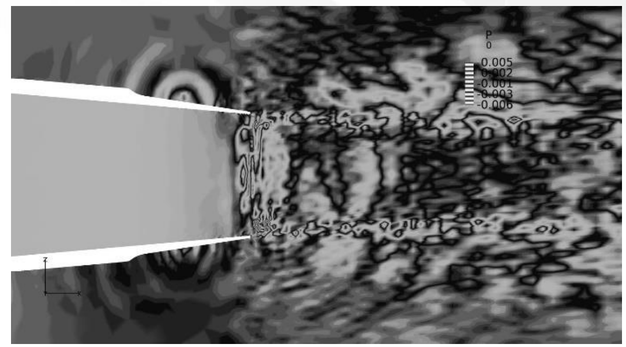

mode 1, modal order 1st



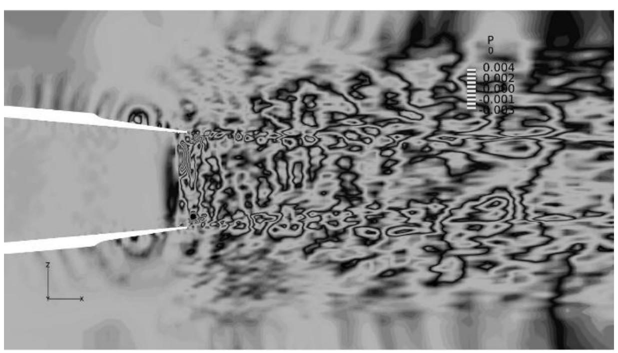

mode 1, modal order 2nd



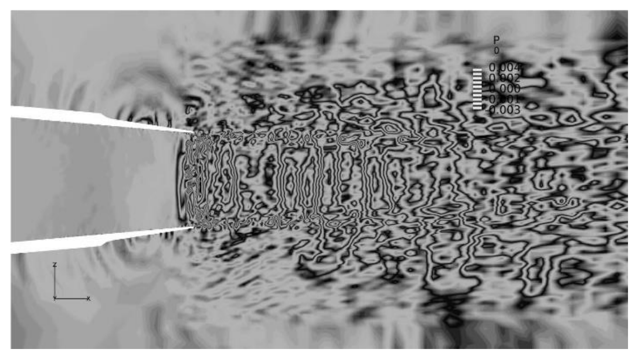
mode 2, modal order 1st



mode 2, modal order 2nd



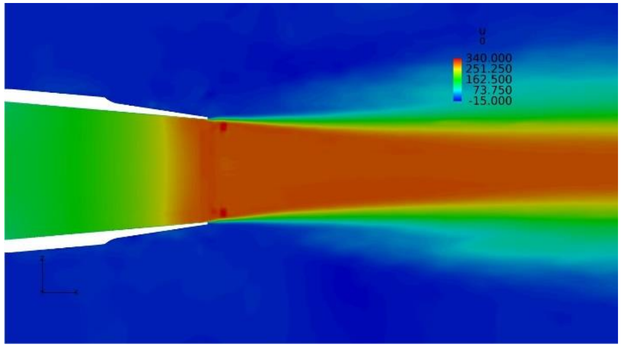
mode 9, modal order 1st



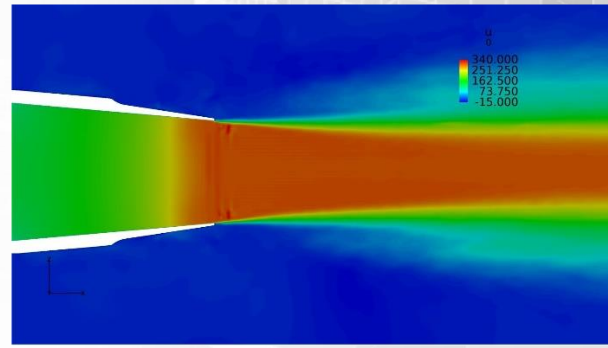
mode 9, modal order 2nd

CAMBRIDGE
FLOW SOLUTIONS

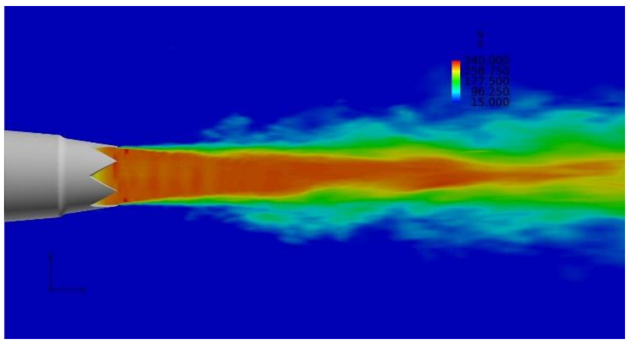
# Reconstruction of velocity field in the X-Z plane (Y=0)



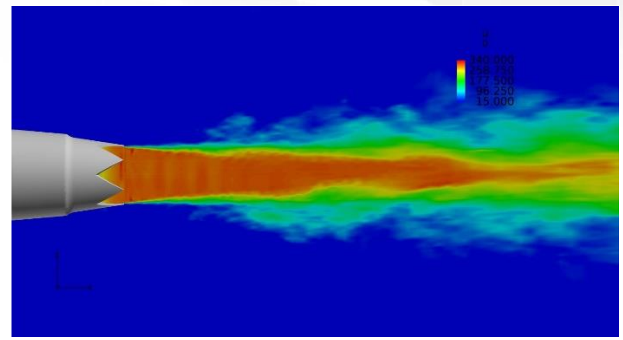

Reconstructed $u$ using mode 1 order 1st



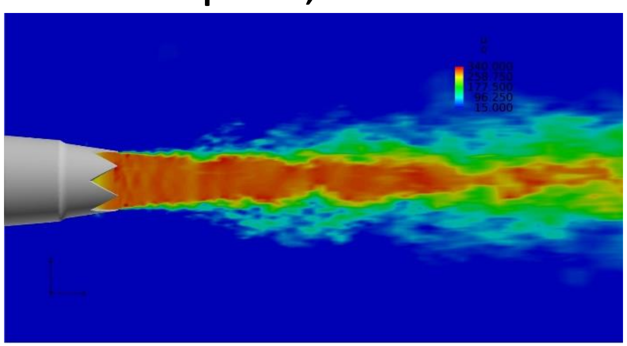

Reconstructed $u$ using mode 1 order 2nd



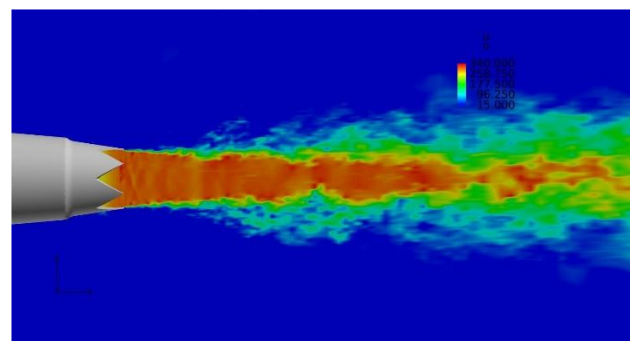
15th snapshots, modal order 1st



15th snapshot, modal order 2nd



145th snapshots, modal order 1st

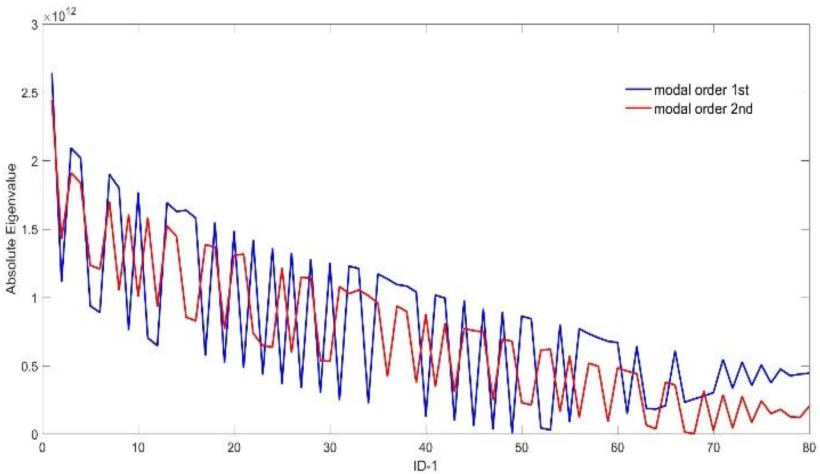


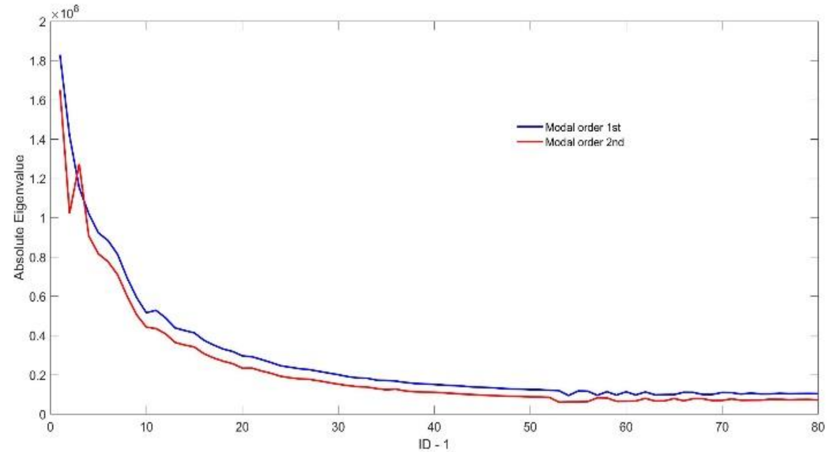
145th snapshot, modal order 2nd

CAMBRIDGE FLOW SOLUTIONS

# HPOD analysis for Landing Gear case

| Job ID | Modal order | No. of variables per cell | Total DOFs | Size of instanenous results (Gb) | No. of snapshops | Size of matrix $A$(single precision) | Actual memory consuming for HPOD (Gb) | Size of modal result for each snapshot (Gb) |
|---|---|---|---|---|---|---|---|---|
| Landing-gear-K0-Pressure | 1st | 1 | 14649682 | 2286.86 | 250 | 13.65 | 20.1 | 0.304 |
| Landing-gear - K1-Pressure | 2nd | 1 | 58598728 | 2286.86 | 250 | 54.6 | 59.86 | 1.216 |
| Landing-gear - K2-Pressure | 3rd | 1 | 146496820 | 457.37 | 100 | 54.6 | | 3.04 |
| Landing-gear - K0-Velocity | 1st | 3 | 43949046 | 2286.86 | 250 | 40.95 | 45.8 | 0.912 |
| Landing-gear - K1-Velocity | 2st | 3 | $1.76 \times 10^8$ | 2286.86 | 250 | 163.8 | 169.1 | 3.648 |
| Landing-gear - K2-Velocity | 3rd | 3 | $4.4 \times 10^8$ | 457.37 | 100 | 163.8 | | 9.12 |

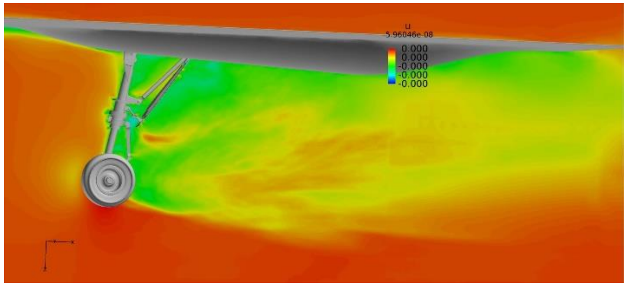


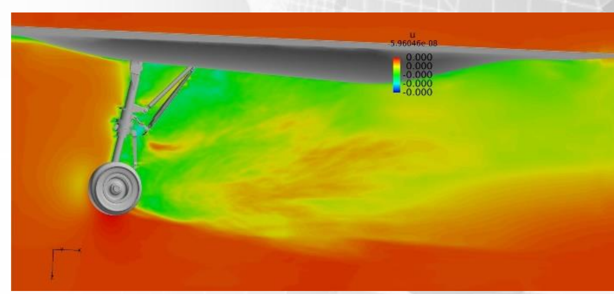

**Plot of Eigenvalues from the HPOD analysis: (left) for pressure and (right) for velocity**

CAMBRIDGE FLOW SOLUTIONS

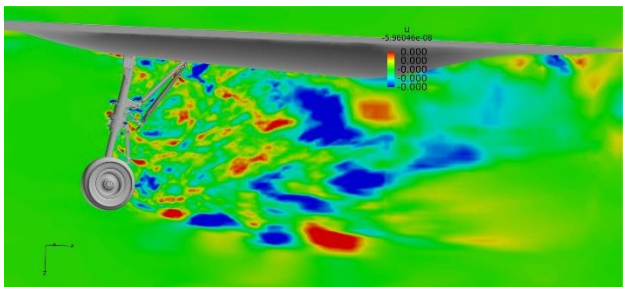
# POD modes in the X-Z plane (Y=0) for velocity u-component



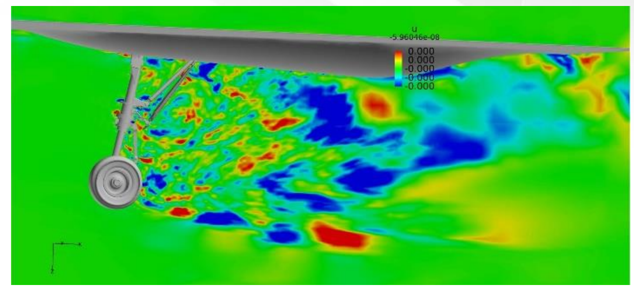
**mode 1, modal order 1st**



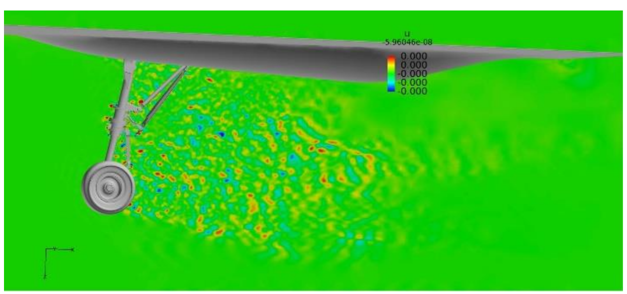
**mode 1, modal order 2nd**



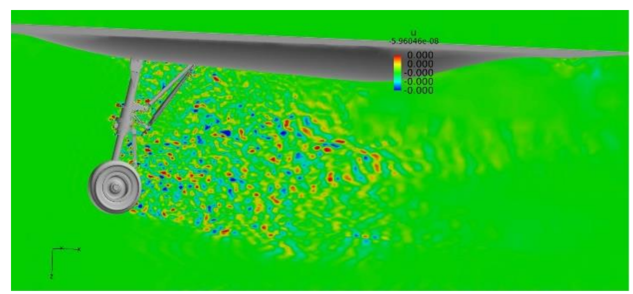
**mode 2, modal order 1st**



**mode 2, modal order 2nd**



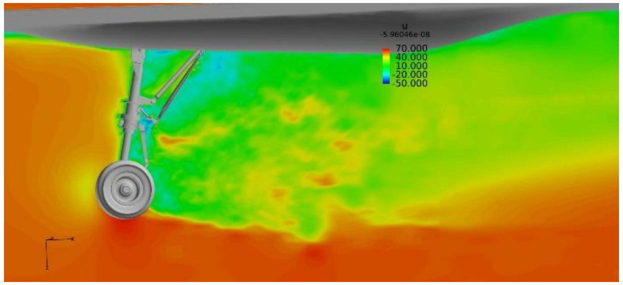
**mode 10, modal order 1st**



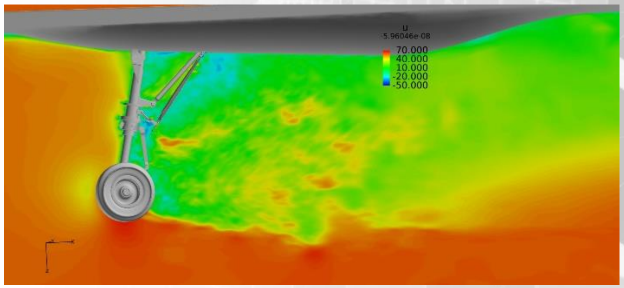
**mode 10, modal order 2nd**

CAMBRIDGE FLOW SOLUTIONS

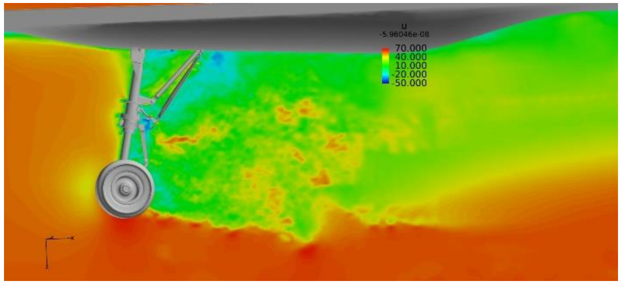
# Reconstructed velocity u-component in the X-Z plane (Y=0)



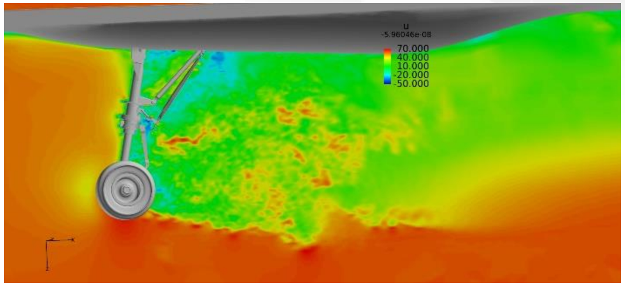

**5th snapshots, modal order 1st**



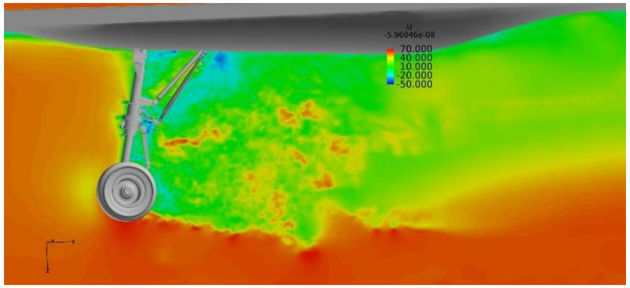

**snapshot, modal order 2nd**



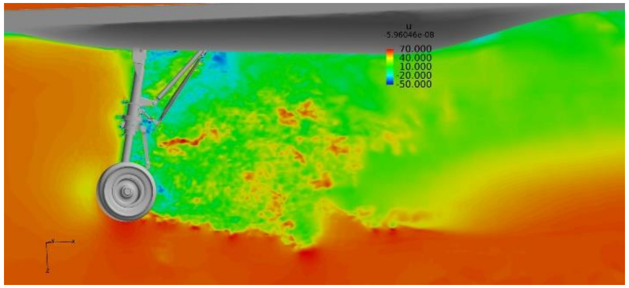

**15th snapshots, modal order 1st**




**15th snapshot, modal order 2nd**




**81th snapshots, modal order 1st**




**81th snapshot, modal order 2nd**

CAMBRIDGE FLOW SOLUTIONS

# Overview

- Paper objective
- Why higher order ?
- The BOXER Environment & LES/*HOTnewt*
- Turbomachinery & aerospace examples
- Computer resources
- Post-processing
- **Summary**

# Summary

- Our objective was *not* to show LES and compare with experiment
  - lots of people do that…
- Our objective is to show ***affordable*** LES which any engineer might perform at their desk
- We have demonstrated LES with acceptable accuracy on some turbomachinery/aerospace applications
- We have demonstrated that useful LES can be performed with significant reductions in computer resource cost
  - the key to unlocking this potential is being able to develop a sufficiently coarse, higher order mesh.
- We have shown the start of our response to the “big data” post-processing challenge in the form of on-the fly POD

CAMBRIDGE FLOW SOLUTIONS

# Appendix

# Higher Order Simulation Methods – PPW

- The key is to understand **mesh resolution**
  - LES is unsteady and needs much greater mesh resolution than that required merely for a steady approximation
  - this resolution requirement increases as the simulation time increases…
  - mesh resolution is measured in terms of ***Points Per Wavelength*** **(PPW)**
- For any numerical scheme the Truncation Error (TE) for convection is of order

$$\Delta x^{p-1}\, \partial^p u / \partial x^p$$

where p depends on the scheme (2 for a P1 method, 3 for a P2 etc.)

- In terms of harmonics $u \sim \Sigma e^{ikx}$ so for a typical wavenumber, $k=2\pi/\lambda$, the order of magnitude of the truncation error, TE, is

$$TE \sim \Delta x^{p-1}\, k^p$$

CAMBRIDGE FLOW SOLUTIONS

# Higher Order Simulation Methods – PPW

- If the simulation takes place over a time $T_{sim}$ then the **total error** for that wave number is of order

$$E_{sim} \sim T_{sim}\, \Delta x^{p-1}\, k^{p}$$

- If $N_{sim}$ is the number of time steps and $\Delta t/\Delta x$ is constant then $T_{sim} \sim N_{sim}\, \Delta x$ then the

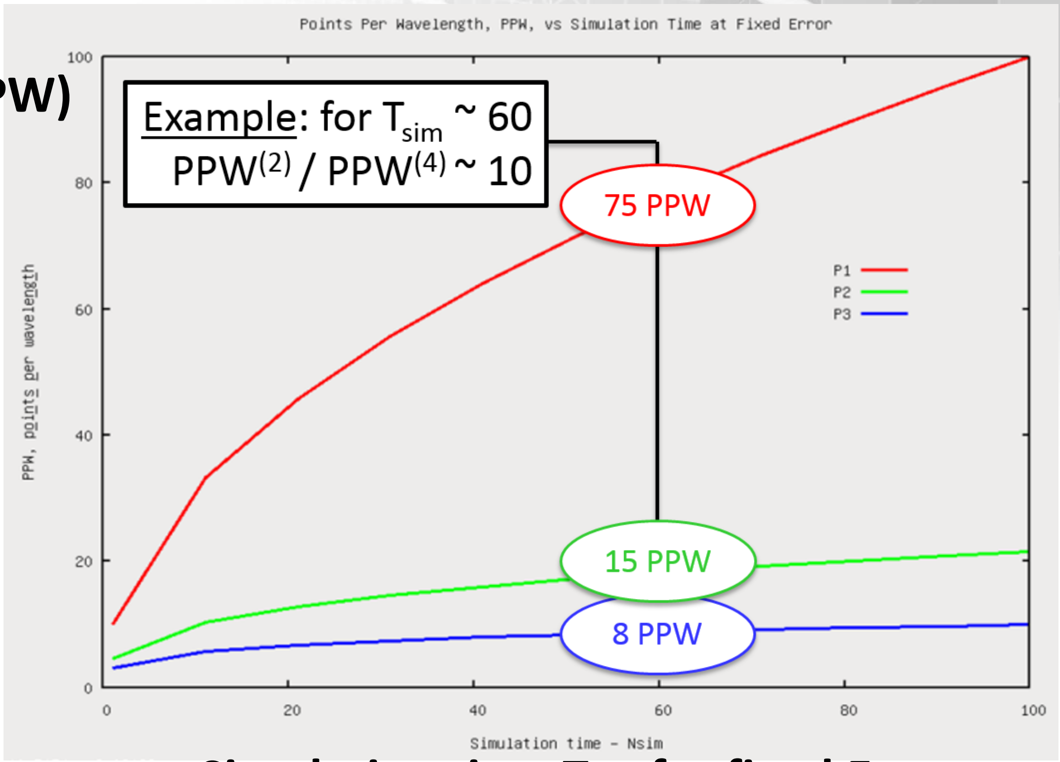
**Points Per Wavelength**, $\lambda/\Delta x$, is given by

$$PPW \sim (N_{sim}/E_{sim})^{1/p}$$

CAMBRIDGE FLOW SOLUTIONS

# Higher Order Simulation Methods – PPW

**Points Per Wavelength (PPW)**




**Simulation time $T_{sim}$ for fixed $E_{sim}$**

- The figure illustrates this trend for P1, P2 & P3…
- This shows that a low order P1 method (“second order”) might need a factor of 10 more PPW in 1D than a P3 method (“fourth order”) – *in 3D this becomes an astonishing factor of 1000 !*
- Since PPW and the associated DOFs (Degrees of Freedom) translate directly to computer memory, RAM, requirement then
  - ***provided the higher order method not consume too much extra memory per DOF then a well-designed algorithm could permit a better computer memory - job size trade-off.***

CAMBRIDGE FLOW SOLUTIONS

# Higher Order Simulation Methods – Floats

- What happens to the operation count – the computer run-time cost?
- Consider a basic _second_ and _fourth_ order convection operator:

$$Q_x^{(2)} = \frac{1}{2\Delta x}(u_{i+1} - u_{i-1})$$

$$Q_x^{(4)} = \frac{2}{3\Delta x}(u_{i+1} - u_{i-1}) - \frac{1}{12\Delta x}(u_{i+2} - u_{i-2})$$

- In terms of operations ***ops$^{(2)}$ = 3*** & ***ops$^{(4)}$ = 7*** if the coefficients (1/2Δx etc.)are re-computed each use (to save memory)

CAMBRIDGE FLOW SOLUTIONS

# Higher Order Simulation Methods – Floats

- So, the *fourth* order method needs more floats BUT for an unsteady simulation, for a given Fourier mode, the *second* order method needs far more Points per Wavelength
  - for example ***$PPW^{(2)} / PPW^{(4)} \sim 10$*** in 1D
- Hence the relative cost in terms of arithmetic operations of a *second* order method compared to a *fourth* order one for the same accuracy in ***3D*** is:

$$floats^{(2)} / floats^{(4)} \sim [(ops^{(2)} / ops^{(4)}) \times (PPW^{(2)} / PPW^{(4)})]^3 \sim 78\ !$$

- ***Thus although the fourth order method needs more operations per DOF this is more than offset by the much reduced number of DOFs for a given accuracy leading potentially to orders of magnitude fewer computer operations – in 3D***
- ***This has strong implications for the energy cost of the simulation***

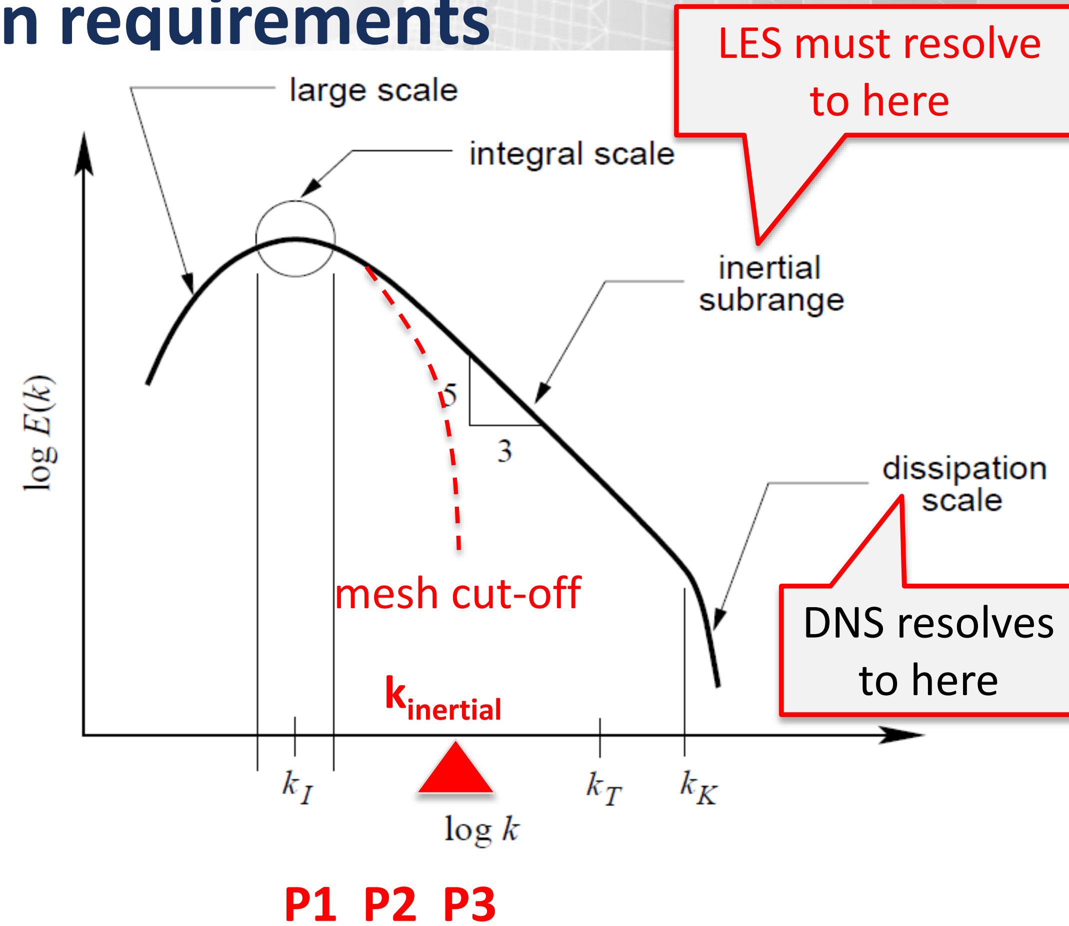
***- roughly - 1 Gflop needs 1 Amp !***

CAMBRIDGE FLOW SOLUTIONS

# Example: LES resolution requirements

- Consider a mesh with **100M** cells – the maximum wavenumber which can be resolved is:

| Order | Max. wavenumber on 100M cells |
|---|---|
| P1 = second | 386. |
| P2 = third | 1932. |
| P3 = fourth | 3623. |




**Only the *fourth* order method can resolve into the inertial subrange !**

**LIABILITY**

Cambridge Flow Solutions Ltd ("The Company") has used all best endeavours to ensure the accuracy of the work performed and any information given, but the Company makes no warranty, express or implied, as to accuracy and will not be liable for any consequences arising out of any inaccuracies or omissions.

The Company excludes all warranties, representations or liabilities to the fullest extent permitted by law. Without prejudice to the generality of the foregoing the Company excludes any liability for consequential loss or damage including economic loss, loss of profit, revenue or goodwill.